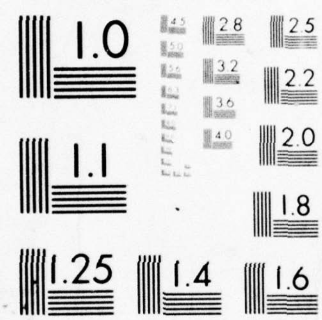
AD-A040 052 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL, (DYNCOM). VOLUME 1. HELICO--ETC(U)
DEC 71 G M CLARK DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-1(U) NL

1 OF 4
AD
A040052



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

FG.

(12)

REPORT RF 2995 FR 71-1 (U)

EXTENSIONS TO THE LAND COMBAT MODEL (DYNCOM) HELICOPTER AND MISSILE MODELS

Final Report

Contract No. DAAH01-70-C-0713

~~This document is subject to Special Export Controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Headquarters, U. S. Army Missile Command, Attn: DRSMI-CM~~

December 31, 1971

by

Gordon M. Clark

SYSTEMS RESEARCH GROUP
Department of Industrial and Systems Engineering
The Ohio State University
Columbus, Ohio 43210

for

SYSTEMS ANALYSIS OFFICE
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (14) RF-2995-FR-71-1 (U)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) EXTENSIONS TO THE LAND COMBAT MODEL, (DYNCOM). HELICOPTER AND MISSILE MODELS. Volume I.		5. TYPE OF REPORT & PERIOD COVERED Final Report, (1/70 to 12/70)
		6. PERFORMING ORG. REPORT NUMBER Jan-Dec 70
7. AUTHOR(s) (10) Gordon M. Clark		8. CONTRACT OR GRANT NUMBER(s) (15) DAAH01-70-C-0713 NEW
9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Research Group, Department of Industrial and Systems Engineering, The Ohio State University Columbus, Ohio, 43210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11) 31 Dec 71
11. CONTROLLING OFFICE NAME AND ADDRESS Systems Analysis Office, DRSMI-CM U. S. Army Missile Command Redstone Arsenal, Alabama 35809		12. REPORT DATE December 31, 1971
		13. NUMBER OF PAGES 376
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 370p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

~~This document is subject to Special Export Controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Headquarters, U. S. Army Missile Command, Attn: DRSMI-CM~~

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Armor	Combat Simulation	Communications	Tactical Doctrine
Helicopters	Computer Simulation	Detection	Environment
Missiles	Programming	Probability	Combat Effectiveness
Crew-Served Weapons	Flow Charts Subroutines	Firepower Land Combat	Protection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This volume describes the design of models developed to represent ballistic-trajectory guided missiles and helicopters in DYNCOM. The missile models are extended to represent missiles flying a ballistic trajectory until acquiring a target and then the tracking of the target is described. The combat actions of helicopters in support of armored units up to battalion size is represented by a complex of models known collectively as The Aerial Platform Combat Operations Module (TAPCOM II). This model is unique in its capability to represent, with high resolution, the movement, target acquisition, (Con't)

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

267355

20. (Continued)

cont. firing, and vulnerability of individual helicopters providing direct aerial fire support. Moreover, the movement and firing tactics of helicopter units are dynamically generated and interactions among the helicopter teams, supported ground force, and other elements of the fire support force are represented.

19. (Continued)

Flight Dynamics
Intelligence
Target Selection
Firing
Lethality
Terrain
Fire Support
Command and Control

ABSTRACT

This volume describes the design of models developed to represent ballistic-trajectory guided missiles and helicopters in DYNCOM. The missile models are extended to represent missiles flying a ballistic trajectory until acquiring a target and then the tracking of the target is described. The combat actions of helicopters in support of armored units up to battalion size is represented by a complex of models known collectively as The Aerial Platform Combat Operations Module (TAPCOM II). This model is unique in its capability to represent, with high resolution, the movement, target acquisition, firing, and vulnerability of individual helicopters providing direct aerial fire support. Moreover, the movement and firing tactics of helicopter units are dynamically generated, and interactions among the helicopter teams, supported ground force, and other elements of the fire support force are represented.

FOREWORD

Research conducted by the Systems Research Group under Contract DAAH01-70-C-0713 with the U. S. Army Missile Command (MICOM), Systems Analysis Office, is reported in three volumes of which this volume is the first. The objective of this research is to extend DYNCOM to represent:

1. helicopter units,
2. ballistic-trajectory guided missiles,
3. movement of crew-served weapons,
4. electronic countermeasures, and
5. effects of smoke and haze on missile guidance systems.

Volume 1, presents the design of models to represent helicopters and ballistic-trajectory guided missiles. Volume 2 contains the DYNCOM program documentation (flow charts and common descriptions) for the helicopters, guided missile, and crew-served weapon models. The Classified Annex, Volume 3, documents the models to represent electronic countermeasures and the effects of smoke and haze.

Conclusions drawn in this report represent the current views of the Systems Research Group, The Ohio State University, and should not be considered as having official MICOM or Department of Army approval, either expressed or implied, until reviewed and evaluated by those agencies and subsequently endorsed.

The cooperation received from MICOM in preparing this report has been extremely helpful. In addition, we wish to acknowledge the advice provided, and the cooperation received, from the U. S. Army Combat Developments Command (USACDC). In particular, the Systems Analysis Group and the aviation agency have been most helpful.

We would like to acknowledge the important contributions of Lois Graber who patiently typed and proofread the text. We also extend our appreciation to the programmers, Robert Wilhelm, Charles McCartney, William Hess, and Gerald Petty, who assisted so ably in developing these extensions to the DYNCOM program. Without their contributions, this model would not be useful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

EXTENSIONS TO DYNCOM - THE LAND COMBAT MODEL

by

D. C. Hutcherson and G. M. Clark

Introduction

This volume, Volume 1, is part of a three volume set reporting several significant extensions to DYNCOM, The Land Combat Model, developed for the U. S. Army Missile Command (MICOM). These extensions consist of the tasks listed below:

1. Integrate the Aerial Platform Combat Operations Model (TAPCOM) into DYNCOM.

2. Develop models to represent the effects of an indirect-fire guided missile in the ballistic trajectory mode.

3. Implement the Crew Served Weapon Movement Model designed under a previous contract.

4. Implement the Electronic Countermeasures Model designed under a previous contract.

5. Develop models to represent the effects of smoke and artificial haze on indirect fire missile guidance systems.

The design of models developed for tasks 1 (Chapters 3 through 9) and 2 (Chapter 2) above are reported in this volume, Volume 1. Program documentation is presented in Volume 2 for tasks 1, 2, and 3. Also, program documentation for tasks 4 and 5 is presented in the classified annex, Volume 3.

The accomplishment of these extensions resulted in significant changes to the structure of DYNCOM. An overview of these changes is presented in this chapter. Also, a review of the history and objectives established for DYNCOM is presented in this chapter so that the reader can appreciate the reasons for the research presented in these three volumes.

DYNCOM, an acronym for DYNamic COmbat Model, is a high resolution Monte Carlo simulation capable of representing engagements between armored forces of battalion size and smaller. These armored forces may employ both conventional and missile armament while being supported by indirect-fire ballistic weapons, by air and ground launched semi-active homing missiles such as MISTIC, and by aerial vehicle units providing direct aerial fire support. DYNCOM is unique in its ability to represent attack, defense and delay operations of ground units while simultaneously representing direct aerial fire support operations of aerial units. Development of DYNCOM has been supported by MICOM; however, many models developed for the U. S. Army Combat Developments Command are used in DYNCOM.

In actuality, DYNCOM is a refinement and an extension of previous simulation models known variously as DYNCOM or DYNTACS. The first version of the model was developed for U. S. Army Combat Developments Command Armor Agency and is described in reference 1. This model was capable of representing engagements between company-sized armor units employing conventional ballistic weapons and was operating successfully on an IBM 7094 computer in March 1967. The model featured a high-resolution treatment of interactions between individual weapon performance, terrain, and environment, and tactical and organizational variables. The model was also designed to permit internal generation of situation dependent tactics and to be flexible in use.

Using this basis, MICOM supported the development of DYNCOM during 1967, 1968, and 1969. Initially, models of the operations supporting-fire maneuver units consisting of indirect-fire MISTIC missile launcher elements were developed. These elements could be ground vehicles or aerial vehicles, but in either case their primary battlefield responsibility was to provide indirect-fire support. These models described the search for targets, time delays in requesting fire, time delays in launching MISTIC weapons, MISTIC missile flight, missile target acquisition, and the missile terminal effects.

Moreover, a model of artillery operations was incorporated. The simulation was expanded to represent battalion sized engagements, and a model of tactical communications was formulated. In addition, a field experiment was conducted at Fort Carson, Colorado to estimate the distribution of message transmission times. All the extensions discussed above are reported in references 2 through 12.

Development of the Aerial Platform Combat Operations Module (TAPCOM) was a significant part of this effort. The primary purpose of TAPCOM was to represent helicopter vehicles that could launch MISTIC missiles and/or request supporting MISTIC fire. TAPCOM incorporated several unique features for a combat operations model of both ground and aerial weapons. The location of individual aerial vehicles was predicted on a continuous basis, and this movement model reflected the unique movement characteristics of helicopters.

Moreover, TAPCOM was designed to represent helicopter operations in support of both attacker and defender forces. Finally, the model included a ground-to-air detection model capable of predicting detection times of sky-lighted target elements.

In late 1969 Combat Developments Command Systems Analysis Group selected DYNCOM as the model to serve as a basis for the development of a simulation suitable for performing the Hard Point Target Study. The simulation model to be developed was to be known as DYNTACS X, and research to achieve the model was to be completed by February 1971.

Analysis of the requirements of the Hard Point Target Study revealed that extension of DYNCOM would be required in three major areas. First, TAPCOM was evaluated for its capability to represent the types of aerial vehicle operations desired in the study. TAPCOM was found to provide a satisfactory basis, but extensions would be required to represent all the desired aerial vehicle activities. The TAPCOM movement model, as it evolved, was very complex and was suitable for representing movement of helicopters performing only those activities specifically represented in TAPCOM. Also, no particular threat to the helicopters existed since air defense weapons were not explicitly represented. Only fire from conventional ground weapons could be employed against helicopters, and the effects of those fires had to be represented by models developed for ground targets. Finally, the main mission for helicopters was an indirect-fire MISTIC launcher mission in which the helicopters operated in a passive manner, launching indirect-fire missiles only upon request. Direct-fire activities were performed only in self defense and even then armament was limited to direct-fire MISTIC missiles. Under certain circumstances a helicopter could also act as an illuminator for indirect-fire missiles launched by other elements.

For the Hard Point Target Study a model that represented much more aggressive helicopter operations was required. Helicopters should be able to perform direct aerial fire support while employing a wide variety of ballistic and missile armament. The model should possess resolution commensurate with previously existing models within DYNCOM and DYNTACS to capture the interactions that exist between individual weapon performance, terrain and environment, and tactical and organizational variables. The model should also represent the interactions that exist between ground and aerial support forces and the coordination of fires that is required when aerial support is employed. Finally, the model should feature internal generation of situation dependent tactics in a manner similar to other DYNCOM/DYNTACS models. Therefore, development of a second generation design for TAPCOM was funded by Systems Analysis Group.

The next area of extension for the Systems Analysis Group was the development of a model to represent air defense weapon operations. The

weapons to be represented included various ballistic and missile systems of the type that might be found on a battlefield of the size used in DYNCOM. Motivation for the model was to provide a capability to perform studies of the performance of helicopters in a hostile environment. It also maintains DYNCOM as a balanced model; that is, each force is provided weapon systems that can counter the threat posed by each weapon system of the opposing force. Finally, it was felt that an unrealistic representation of artillery operations existed in DYNCOM. The problem was that no counterbattery operations were represented. Artillery could inflict damage upon the ground forces on the battlefield but no means existed to degrade the performance of the artillery. Therefore, a model of artillery counterbattery operations was funded for development. The results of the research for the System Analysis Group to develop the extensions listed above are reported in reference 13, 14, and 15.

Because of concurrent contracts with MICOM to integrate TAPCOM into DYNCOM and to extend DYNCOM for the Systems Analysis Group, an integrated model was developed that would simultaneously satisfy MICOM and Systems Analysis Group requirements. This report describes the resulting design for TAPCOM II in Chapters 3 through 9. The reader is referred to references 13, 14, and 15 for a description of the counterbattery fire models, air defense models, and fire support coordination models that are fully compatible with DYNCOM as described herein.

As a result of this integrated effort, the final DYNCOM simulation has sufficient flexibility to represent combat engagements between armored forces consisting of both ground and aerial vehicles ranging in size up to an armored battalion. A coordinated attack against a sequence of objectives conducted by several teams moving within separate axes of advance can be represented. As this maneuver force advances, a base of fire is provided by supporting-fire units on the ground and in the air. The supporting fire units on the ground can move forward to fire support positions and thus cover the entire advance. The aerial units can move quickly about the battlefield and provide direct aerial fire support as needed.

On the defender side multiple maneuver units may also be represented and each can be assigned to a sequence of delay positions. Supporting fire from ground and aerial units can also be provided for this delaying force so that the entire withdrawal can be covered. Moreover, artillery fires against advancing and withdrawing units can be represented for the duration of the battle. Thus, DYNCOM represents a dynamic combat situation in which all opposing forces can be moving simultaneously.

Introduction of Helicopter Operations

Introduction of helicopter operations of the type required in DYNCOM has led not only to the development of an extensive complex of models dealing with combat activities of helicopters but also to an extensive revision of other portions of the simulation. These latter revisions have been necessitated by the need to provide a highly centralized method of controlling not only the fires of helicopters but also the fires of other support type weapons as well. To introduce the material we must first understand the general role to be played by aerial vehicles in an armored engagement. We must then ascertain the ways in which the operations of aerial vehicles are controlled during an engagement. We will then be able to understand the need for an integrated fire support model within DYNCOM.

The Role of Aerial Vehicle Units

The type of aerial vehicle selected for simulation is the army helicopter. Thus, we exclude from analysis all Army and Air Force fixed-wing aircraft. The reader should note that the types of missions to be discussed emphasize the aerial vehicle's role during fire support missions which utilize direct or indirect fire. Thus, army helicopters capable only of reconnaissance, troop transport or command functions would be poor choices for vehicles to be represented. The simulation would accept performance descriptions of these types of vehicles, however.

Our only emphasis is upon the army helicopter in aggressive support of armor in a small-unit engagement. Thus, the mission of the simulated aerial vehicle units is to deliver direct aerial fire support. According to reference 6, direct aerial fire support is

> . . . fire support delivered by aircraft organic to ground forces against surface targets and in support of land operations. Coordination by the attack helicopter commander and the ground commander allows helicopter fires to supplement and be integrated into the committed firepower of the ground force.

Thus, we simulate aerial vehicle units operating in close coordination with the supported armor unit. A question that remains is whether this support is <u>preplanned</u> or <u>immediate.</u>

Again, from reference 6

> Preplanned fires are those that are planned for delivery in advance of takeoff. These fires are closely coordinated with the ground force commander and his fire support coordinator to insure support of the ground tactical plan. Planning normally includes target location, type

and amount of weapons and ammunition, time of delivery, technique of delivery and method of adjustment.

On the other hand, immediate direct aerial fire support fires are used against targets of opportunity, or they become necessary because of changes in the tactical situation.

Immediate fire targets may be acquired by an individual or element in the battle area; however, within his area, the ground commander is responsible for the control of these fires. All immediate fires require close coordination of the fire team leader and the ground commander or his fire support coordinator.

Within DYNCOM, both preplanned and immediate fires may be simulated. As will be discussed, it is possible to simulate all of the preplanned missions: preparation fires, harassing fires, diversionary fires, interdicting fires, and counterpreparation fires. However, simulation of the delivery of immediate support is more restricted. The models are designed primarily to simulate aerial units acting as part of a base of fire. That is, aerial units deliver immediate direct aerial fire support in response to fire requests that are generated as the engagement proceeds. Normally, this characteristic of the model would require that there be at least some elements on the ground to be supported. However, this limitation is not unduly restrictive. For example, should the user desire that one of the opposing forces be composed entirely of aerial units, input data could be prepared so that aerial units would perform preplanned missions to initiate the action. Once initiated, the action would proceed with aerial units attacking targets of opportunity or delivering area countermeasure fires. The reader should note, however, that hostile aircraft are not considered for attack in the simulation. Thus, at least one force must have some ground elements. Also, countermeasure fires will be delivered only against ground weapons posing a threat to the aerial vehicles. In a subsequent section of this chapter, we will discuss the missions that are represented in DYNCOM so that the various types of helicopter fires discussed may be simulated.

While performing either preplanned or immediate support missions within DYNCOM, aerial vehicle teams may deliver neutralization fires, destruction fires, or combined fires. The first of these fires are intended to reduce the combat efficiency of the enemy by hampering the fire of his weapons, reducing his freedom of action and movement, and reducing his ability to inflict casualties. Destruction fires are delivered for the sole purpose of destroying enemy troops and equipment. Finally, an aerial vehicle carrying a mixed ordnance load may deliver combined fires. That is, neutralization fires may be delivered for protection while engaging a point target with destruction fires.

Fire-Support Coordinator

As stated above, the simulated aerial vehicle mission is either preplanned or immediate direct aerial fire support. Thus, according to doctrine, the aerial vehicle units in support must coordinate their activities with all other units participating in the engagement. In actual practice, this coordination is brought about by positive command and control measures exercised by the force commander (or his fire-support coordinator). Therefore, in DYNCOM, target selection and mission assignment procedures for aerial vehicles must carefully account for coordination among a variety of weapon types. Not only must the aerial vehicles perform in an integrated fashion with elements of the maneuver force, but they must also supplement the fires of other types of supporting weapons. These weapons include conventional tube-type artillery, indirect-fire ballistic rockets, mortars, naval gunfire, and indirect-fire semi-active missiles (MISTIC).

The only way in which coordination can be satisfactorily accomplished in DYNCOM is through a highly centralized and integrated target selection and mission assignment procedure. Unfortunately, this fact has resulted in a considerable increase in scope of effort compared with initial concepts contemplated at the time the initial development of TAPCOM was contemplated. At that time, it was thought that coordination features could simply be incorporated as part of the design of the aerial vehicle simulation. However, during the course of research, it was determined that separate coordination features were unsatisfactory. The evolutionary nature of DYNCOM has resulted in a very fragmented approach to representing coordination, and another new model would only compound the deficiency. For example, coordination of fires between elements of the maneuver force and those of the MISTIC supporting force was treated by the Fire Controller reported in reference 7. However, coordination of fires among elements of supporting artillery was treated by the Artillery Model described in reference 3. No method existed for coordinating fires between artillery and MISTIC units.

Centralized Modeling Approach

In order to develop a centralized procedure for describing target selection and mission assignment processes, it was necessary to analyze the fire request processes involved with each type of supporting weapon system, with the hope that the processes were similar. As was hoped, investigation revealed that all processes were essentially the same, once appropriate analogies were developed. Thus, one logical structure was found to be sufficient to handle all the supporting systems. We shall illustrate the required analogies by describing the fire-request structure associated with artillery, and the similarities between this structure and that of MISTIC and Army air.

In artillery operations, a line-of-sight relationship normally does not exist between the target and an artillery unit assigned to fire on the target. The target is detected and identified by an observer who requests fire on the target through a fire direction center. The fire direction center converts the requests into firing data and assigns the fire mission to a battery. The battery then applies the data to the weapons and executes the mission. All the while, the force commander and his fire-support coordinator are monitoring the operation to insure conflicts do not arise.

In the case of MISTIC weapons, a similar procedure is followed. Again, a line-of-sight relationship between target and missile launcher usually does not exist. Thus, a forward observer equipped with an illuminator detects and identifies suitable targets. These fire requests may be addressed directly to one of the missile launchers, but it is also possible that a facility similar to an artillery fire direction center might be used. In any event, the request is processed by some combination of agents to convert the request into firing data that may be applied to the missile launcher for execution. Again, the fire-support coordinator monitors fire requests to resolve conflicts.

In the case of attack helicopters performing immediate direct aerial fire support, targets are assigned to helicopter teams (analogous to artillery firing batteries or MISTIC launchers) by some member of the supported unit's *command team (force commander,* fire-support coordinator, etc.). This command team is analogous to the artillery fire direction center in terms of target assignment operations.

The target could have been identified by almost anyone participating in the engagement. The commander may have called for reconnaissance by fire on a suspected target location, or some member of the maneuver force may have requested supporting fire on a detected enemy target complex. In any event, the element that identifies the target and requests the fire is acting in a manner analogous to the artillery forward observer.

Thus, we see that the components of the fire-request structure that exist for initiating artillery fire support have their counterparts in the structure for initiating MISTIC and Army air support fires. Similar procedures may be used throughout.

We have, however, neglected to discuss fires delivered on targets of opportunity which were discovered by the supporting weapons themselves. These targets are discovered in a fashion completely unrelated to the processes discussed above for the indirect-fire problem. Thus, target selection and assignment procedures in these cases are quite different and require an entirely different modeling approach. This may be accomplished by having, in addition to a centralized fire-support target selection procedure, other fire control models which assess the desirability of direct fire on targets of opportunity.

The fact is that such models have been designed for DYNCOM. The Fire Controller of reference 7 is designed to permit selection of direct-fire targets by MISTIC launchers in self defense. Moreover, the TAPCOM II mission controller of Chapter 4 permits initiation of countermeasure missions against targets discovered by aerial teams. Such activities preempt normal fire-support functions if need be.

We will now turn our attention to a discussion of some of the important concepts used in DYNCOM and especially in the fire-support model that has been developed.

Elements and Organizations

A discussion of elements, events and organizations assumed in DYNCOM is presented in the following paragraphs. This discussion will serve to make explicit the relationships between the fire-support model and other models of DYNCOM.

Elements

DYNCOM has always employed the concept of an element to represent combat. Before, an element was the smallest combat entity provided a point location on the battlefield and might be a tank, an armored personnel carrier, or an antitank crew-served weapon. In DYNCOM, we call such elements vehicular elements because we have expanded the concept of an element somewhat. Now we have a wide variety of special fire-support elements that are not even necessarily associated with a vehicle.

Fire-support elements in DYNCOM are all those elements on the battlefield that have anything whatsoever to do with requesting, controlling, or executing fires from indirect-fire ballistic weapons, MISTIC launchers, or attack helicopters. However, an element does not necessarily consist of one individual. More commonly, the element will consist of several individuals working as a team with respect to some fire-support task. For example, in DYNCOM, fire-support missions are assigned to attack helicopter teams and artillery firing batteries. Each of these DYNCOM fire-support elements actually consist of several entities, viz., artillery tubes and attack helicopters. As another example, the fire-support coordinator element actually represents various individuals working together. The element might consist of the force commander, the fire-support coordinator, the operations officer, the operations officer for air, and the intelligence officer. In Table 1.1, we have summarized several of the fire-support elements represented in the DYNCOM fire-support model. We will discuss each of these elements in more detail as we discuss fire-support organization. For a more detailed discussion of vehicular elements, see Chapter 1 of reference 2.

Table 1.1

DYNCOM Fire-Support Elements

DYNCOM Element	Combatants Represented
Aerial Team	Light or heavy attack helicopter team
Artillery Firing Battery	Artillery or mortar firing battery
MISTIC Launcher	MISTIC launcher and crew
Artillery Fire Direction Center	Artillery or mortar firing battery fire direction center
MISTIC Unit Fire Controller	Command and control team within a MISTIC unit
Forward Observer	Artillery, mortar, or MISTIC forward observer team, or individual element who needs fire support
Fire Support Coordinator	Battalion command and control team
Artillery Intelligence Center	Counterbattery operations command and control

Organization

Tactical organizations of vehicular elements within DYNCOM are constructed by forming maneuver units composed of elements with different capabilities. With the exception of aerial vehicles, the movement activities of individual elements are controlled by these maneuver units. The method of control for aerial elements is discussed in a subsequent section. While moving, ground vehicles guide on their maneuver unit leaders who designate routes and formations. If a maneuver unit is occupying a stationary position, the decision to initiate movement is made by the maneuver unit leader.

A maneuver unit can vary in size from one to seven platoons, and a maneuver unit composed of more than one platoon is a team maneuver unit. Platoons can have as many as two sections, and they can be as small as a single element. Sections vary in size from one to four elements.

The organization of elements on the battlefield can affect the flow of intelligence information concerning the location of enemy weapons. Each platoon is provided a communication net known as the platoon tactical net. Then, there are several company tactical nets to which platoon leaders and others might be assigned. Finally, there is a battalion tactical net for each side. The elements on each net are established by input data with net organization altered when casualties occur. It is over these nets that enemy intelligence information is passed. Fire-support communications employ a different net structure which will be outlined below.

It is assumed that all the fire-support elements discussed previously are also organized into units. This organization structure may be controlled by input data to parallel structures found in actual combat units.

An artillery unit consists of one firing battery, one fire direction center, and possibly several forward observer teams belonging to the unit and assigned to support the battalion operation. A forward observer team numbered NFO belongs to the unit specified by INART(NFO). INART is a linear array with one entry for each observer team being simulated. The array must be input, and zeros are used to indicate forward observer teams which are not members of artillery units. There are NUMART artillery units, with NBART of these units being elements of the blue force.

A MISTIC unit is very similar in organization to an artillery unit. It consists of several launchers, a command and control team, and several forward observer teams assigned to support the battalion operation. There are MISTUN units simulated, with NBMIS units assigned to support the blue force.

An observer team numbered NFO belongs to MISTIC unit IMIST(NFO). This array must be initialized for each observer. Again zeros are used to indicate FO teams which are not MISTIC forward observers.

Launchers are assigned to MISTIC units through the array NMISUN(LNC). This array must be initialized for all launchers LNC. In general, there are ITOTLN launchers on the battlefield, and each launcher LNC fires missions for unit NMISUN(LNC).

An aerial vehicle unit consists of several attack helicopters operating together as a team. There are NUMAVT teams simulated, with NBAVT teams assigned to support the blue force. Missions for these teams are first assigned by the fire-support coordinator element.

In general, each side (red or blue) is allowed only NTOBAL(KP1) aerial teams under active control over the battlefield at any time,where KP1 = 1 for the blue force and KP1 = 2 for the red force. The activities of these teams will be discussed in more detail later,and the meaning of being under active control and over the battlefield will become clearer then.

Types of Forward Observer Teams

In general, there are two classes of forward observer teams operating on the battlefield. The first class consists of teams that we call regular FO's. These teams correspond to those that are actually assigned by an artillery or MISTIC unit to support battalion operations. Thus, one way to determine whether forward observer NFO is a regular FO is for either INART(NFO) or IMIST(NFO) to be positive. Note that it is possible to specify that a regular FO is a member of both an artillery and a MISTIC unit. However, the model is designed to treat the FO more as an artillery FO than as a MISTIC FO when such is the case. The significance of being a regular FO is clarified by the discussion of the types of targets for which an FO may request fire in Chapter 2 of reference 15.

Another way to determine whether forward observer team NFO is a regular FO is to determine that it is not a special FO. A special FO is one that belongs neither to a MISTIC unit nor to an artillery unit. Thus, if NFO is a special FO, INART(NFO) = IMIST(NFO) = 0. The special FO represents an element on the battlefield that may desire to receive assistance during an intense fire fight with elements of the enemy force. The assistance is obtained by requesting fire support to be delivered for the purpose of neutralizing or destroying the immediate enemy threat. The special FO may be any member of the maneuver force, but in practice the special FO would normally be at least a section or platoon leader. Special FO's are discussed in more detail in the presentation of the types of targets for which an FO may request fire in Chapter 2 of reference 15.

For convenience, all FO teams are numbered so that the first NTSFO teams are special FO teams. Thus, within the simulation, to determine whether or not FO team NFO is a special team, it is necessary only to compare NFO against NTSFO. That is, if NFO $\leq$ NTSFO, NFO is a special FO team; and, if NFO > NTSFO, NFO is a regular FO team. This convention requires that the user input INART(NFO) = IMIST(NFO) = 0 for special FO teams NFO.

Correspondence Between Fire Support Elements and Vehicular Elements

The reader should note that FO teams are not regular DYNCOM vehicular elements. That is, FO teams are processed by a special portion of the DYNCOM main program that deals only with FO target selection activities. On the other hand, vehicular elements are elements such as tanks, armored personnel carriers and individual attack helicopters. These elements are processed through models designed to represent the accumulation of intelligence about the enemy, the decisions required for firing and movement, and the actual fire and movement that results from the decisions made. These elements are not processed through any portion of the FO model.

The question then is how do we represent the movement of FO teams and the accumulation of enemy intelligence required to develop fire requests. The answer is that the FO teams are assumed to be transported by regular vehicular elements. The intelligence possessed by the FO teams is that which is possessed by the crews of the vehicles carrying them. Likewise, movement of an FO team is simply the movement of the transporting vehicle. In short, the tactical situation existing for an FO team is identical to that of the transporting vehicle. A further discussion of FO activities appears in Chapter 2 of reference 15.

Concepts similar to those above also apply to the MISTIC launcher fire-support element. MISTIC launcher vehicles are processed by the regular DYNCOM models designed to represent accumulation of intelligence, the decisions required for firing and movement, and the actual fire and movement that results from the decisions made. However, firing decisions and firing activities for these vehicles are usually restricted to be only those direct-fire activities required for self defense.[1] In the event that self-defense firing

---

[1]The model is actually designed to accept any direct-fire tactic specified by the user. However, it is assumed that the normal tactic would stress direct fire as a self-defense mechanism. Consequently, direct fire takes precedence over indirect fire in the model. Thus, it is possible that an ongoing indirect-fire mission might be interrupted by a direct-fire assignment.

activities are not required, then the MISTIC launcher fire-support element is activated to perform indirect-fire MISTIC launcher activities. These activities are represented in a special portion of the DYNCOM program and include all required communications with forward observers requesting and verifying fire missions by MISTIC launchers. The MISTIC launcher fire-support element is described in Chapter 3. The loading of indirect-fire MISTIC missiles and firing a sequence of MISTIC missiles against a target located by a forward observer are represented by the launcher vehicular element. The direct and indirect firing activities of MISTIC ground launchers are presented in Chapter 3. Chapters 6 and 8 present helicopter MISTIC launcher operations.

Correspondence between FO teams and the vehicles transporting them is established in the array NOBVH. This same array is also used to establish the correspondence between MISTIC launcher fire-support elements and MISTIC launcher vehicles. NOBVH is a linear array with one entry for each FO team followed by one entry for each MISTIC launcher fire-support element. It must be initialized with positive integers corresponding to the element number of the proper vehicular elements. In general, the elements may correspond to tanks, APC's, or even attack helicopters. However, under certain conditions, element numbers of zero may be entered. Such a feature is required to permit the representation of FO Bravos as discussed in Chapter 2 of reference 15 and to allow for creation of FO teams and launcher elements during the conduct of an engagement as discussed in Chapter 6.

The aerial units referred to earlier as fire-support elements are related in a definite way to maneuver units consisting of vehicular elements. The fire-support element NAT for an aerial maneuver unit is the decision element that controls the overall operations of the maneuver unit. This element is processed through a special portion of the DYNCOM main program and its function is to make decisions as to what overall mission activities are to be performed by the aerial unit. All required communications with other elements of the fire-support force are also conducted by NAT. Elements within the maneuver unit then form movement and firing decisions within the framework provided by the decisions of NAT.

The correspondence between NAT and its associated aerial maneuver unit is established by the initialized arrays KMANU and MANHEL. The first of these arrays gives the vehicular maneuver unit corresponding to each aerial unit decision element, while MANHEL gives the aerial decision element associated with each vehicular maneuver unit. Obviously, a zero should be entered in MANHEL for each maneuver unit that is not an aerial maneuver unit. The reader can visualize NAT as the leader of an aerial maneuver unit if he so desires. For a further discussion of the decision activities, see Chapter 4.

Two fire-support coordinator elements for each side (red and blue) are actually employed within DYNCOM. As specified in Table 1.1, these elements correspond to the battalion command and control team, but there is

no actual association with a particular vehicle on the battlefield. These elements simply perform the decision making activities that such teams would be called upon to perform. Each is processed through a special portion of the DYNCOM main program.

The first fire-support coordinator element (FSC) monitors fire request communications from forward observers and attempts to resolve any conflicts that arise during the course of the battle. This element commands by exception in this mode. However, the element also operates actively when called upon to do so. An example of this mode of operation occurs when a forward observer requests aerial fire support. Then, the FSC attempts to locate an available aerial team for assignment to the mission. These activities are explained in further detail in Chapter 2 of reference 15.

The second FSC element is also referred to as the ground-to-air communicator or GAC. This element is responsible for determining appropriate targets for aerial teams and requesting support when targets are found. As will be discussed in Chapter 2 of reference 15, the GAC determines when aerial teams should investigate enemy strong points or act as forward observers or indirect-fire MISTIC launchers.

The artillery intelligence center element is responsible for controlling artillery counterbattery fires as reported in reference 13. There is one of these elements per side and no association exists with vehicular elements or maneuver units. The element is processed through a special portion of the DYNCOM main program and generates counterbattery fires, either from artillery units or from aerial units. Operations of the element are further discussed in Chapter 4 of reference 15.

The operations of artillery firing batteries are discussed in detail in reference 3, and, as modified for counterbattery operations, they are described in reference 13 and Chapter 4 of reference 15. As stated previously, there is one battery per artillery unit and each battery is capable of firing scheduled fires, triggered on-call fires, target-of-opportunity fires and counterbattery fires. These operations are simulated in a special portion of the DYNCOM main program, and, of course, no association exists between an artillery firing battery and any vehicular element. The guns of a battery are assumed fixed in location.

The final fire-support elements are those associated with the fire-request radio net organization constructed for the fire-support force. There is one net per artillery unit, one net per MISTIC unit and two special purpose nets per side. The first special purpose net is a ground-to-ground net used by forward observers requesting aid from the fire-support coordinator. The

other net is a ground-to-air net used for all fire-request communications between ground elements and aerial elements. The radio net elements have been created to represent the activity states of each net (open or closed) as a function of time, and these elements are processed through a special portion of the DYNCOM main program. The radio net elements are also referred to as fire direction center elements since a data processing function is also simulated when a net element is active. No association exists between the net elements and any vehicular element. Operations of the net elements are described in detail in Chapter 3 of reference 15.

Numbering of DYNCOM Elements

With the multitude of element types described above, it is obvious that we must have an element numbering scheme so that they may be identified for processing. An orderly numbering scheme also permits a more orderly processing scheme.

In Table 1.2, we have summarized the element numbering scheme used in DYNCOM. There, we have indicated the classes and subclasses of elements simulated and the element numbers that are allowed within each class and subclass. Element classes 4 through 7 are those of principal interest in the fire-support model. Class 4 elements represent both the regular and special FO teams that are simulated. Class 5 elements are the fire control elements and radio net elements of interest, while class 6 elements are the firing units which execute the missions assigned by class 5 elements. Note that some of the class 5 elements represent the fire-support coordinator elements on the battlefield while the other class 5 elements are associated with radio nets that were discussed. The class 7 elements are the artillery intelligence center elements of the counterbattery fire model.

The transporting vehicles discussed previously are contained within element class 1. That is, the maximum value that may appear in the NOBVH array is NUMELE, as this is the number of the last regular vehicular element.

The cumulative element number appearing in the right-most column of Table 1.2 is called the clock number. The reason for this convention is that each simulated element has a variable ECLOCK(I) associated with it. This variable is the battle time at which the last event for element I was completed. Thus, within DYNCOM, the array ECLOCK may be used to determine the order in which elements should be processed. That is, the next element to be processed at the end of any given event is that element ICE for which

$$\mathrm{ECLOCK(ICE)} = \min_{1 \leq I \leq \mathrm{NCLK}} (\mathrm{ECLOCK(I)}).$$

Table 1.2.--Element Numbering Summary

Class Number	Class Description	Sub Class Description	Example	Element Number Within Sub Class	Element Number Within Class	Clock Number
1	Vehicles	Blue	Tank, APC, Attack Helicopter, Crew Served Weapon, Air Defense Weapon	1 ⋮ NUMBLU	1 ⋮ NUMBLU	1 ⋮ NUMBLU
		Red		1 ⋮ NUMELE-NUMBLU	NUMBLU + 1 ⋮ NUMELE	NUMBLU + 1 ⋮ NUMELE
2	Output	--	Used to control periodic output of summaries	--	1 ⋮ NMSCLK	NUMELE + 1 ⋮ NUMELE+NMSCLK
3	MISTIC Missiles	--	Used when a MISTIC missile is created	--	1 ⋮ MNMIS	NUMELE+NMSCLK+1 ⋮ NTFO
4	Forward Observers	Special	Section leader or platoon leader	1 ⋮ NTSFO	1 ⋮ NTSFO	NTFO + 1 ⋮ NTFO+NTSFO
		Regular	MISTIC or Artillery unit FO	1 ⋮ ITOTFO-NTSFO	NTSFO + 1 ⋮ ITOTFO	NTFO+NTSFO+1 ⋮ NTFDC
5	Fire Control Elements	Blue Artillery Fire Direction Centers	Indirect fire ballistic weapon fire control center	1 ⋮ NBART	1 ⋮ NBART	NTFDC + 1 ⋮ NTFDC+NBART
		Red Artillery Fire Direction Centers		1 ⋮ NUMART-NBART	NBART + 1 ⋮ NUMART	NTFDC+NBART+1 ⋮ NTFDC+NUMART
		Blue MISTIC Fire Direction Centers	Indirect fire semi-active homing missile fire control center	1 ⋮ NBMIS	NUMART + 1 ⋮ NUMART+NBMIS	NTFDC+NUMART+1 ⋮ NTFDC+NUMART+NBMIS
		Red MISTIC Fire Direction Centers		1 ⋮ MISTUN-NBMIS	NUMART+NBMIS+1 ⋮ NUMART+MISTUN	NTFDC+NUMART+NBMIS+1 ⋮ NTFDC+NUMART+MISTUN
		Blue Fire Support Coordinator	Force commander, operations officer, fire support coordinator, intelligence officer	1	NUMART+MISTUN+1	NTFDC+NUMART+MISTUN+1
		Red Fire Support Coordinator		1	NUMART+MISTUN+2	NTFDC+NUMART+MISTUN+2
		Blue battlefield Command Team	Force commander, operations officer, fire support coordinator, intelligence officer	1	NUMART+MISTUN+3	NTFDC+NUMART+MISTUN+3
		Red battlefield Command Team		1	NUMART+MISTUN+4	NTFDC+NUMART+MISTUN+4
				1	NUMART+MISTUN+5	NTFDC+NUMART+MISTUN+5
		Blue Air to Ground Net	Air to ground radio net used for communication with aerial teams			
		Red Air to Ground Net		1	NUMART+MISTUN+6	NTFB
6	Firing Batteries	Blue Artillery Firing Batteries	Indirect-fire ballistic weapon firing unit	1 ⋮ NBART	1 ⋮ NBART	NTFB + 1 ⋮ NTFB+NBART
		Red Artillery Firing Batteries		1 ⋮ NUMART-NBART	NBART + 1 ⋮ NUMART	NTFB+NBART+1 ⋮ NTFB+NUMART
		Blue MISTIC Launcher Crews	Indirect fire semi-active homing missile launcher	1 ⋮ NBLULN	NUMART + 1 ⋮ NUMART+NBLULN	NTFB+NUMART+1 ⋮ NTFB+NUMART+NBLULN
		Red MISTIC Launcher Crews		1 ⋮ ITOTLN-NBLULN	NUMART+NBLULN+1 ⋮ NUMART+ITOTLN	NTFB+NUMART+NBLULN+1 ⋮ NTFB+NUMART+ITOTLN
		Blue Aerial Vehicle Teams	Attack helicopter team decision element	1 ⋮ NBAVT	NUMART+ITOTLN+1 ⋮ NUMART+ITOTLN+NBAVT	NTFB+NUMART+ITOTLN+1 ⋮ NTFB+NUMART+ITOTLN+NBAVT
		Red Aerial Vehicle Teams		1 ⋮ NUMAVT-NBAVT	NUMART+ITOTLN +NBAVT+1 ⋮ NUMART+ITOTLN +NUMAVT	NTFB+NUMART+ITOTLN+NBAVT+1 ⋮ NAIC
7	Artillery Intelligence Centers	Blue AIC	Decision element for counterbattery operations	1	1	NAIC + 1
		Red AIC		1	2	NCLK

The element ICE is called the current element and NCLK is the total number of elements being simulated. The meaning of the term event will be explained in more detail in a subsequent paragraph.

All named variables appearing in the right-hand columns of Table 1.2 are input to the simulation. For an explicit definition of these variables, see the descriptions of COMMON/NUMBER/, COMMON/SEQPAR/ and COMMON /NTELE/ in Volume 2.

Events

DYNCOM simulates a battle by specifying a sequence of activities for each element for the duration of the engagement. The interactions among the activities of each element are resolved by defining fundamental events with respect to its actions. An event is a commitment to action during which an element will not alter its activities regardless of the actions of other elements. For a ground vehicular element examples are provided by movement for a short time interval, firing one main gun round, or a single burst from a rapid-fire weapon. The nature of each event implies a time to perform the event which can be specified deterministically or stochastically. We will discuss this phenomenon in more detail as we discuss events for the various types of DYNCOM elements in the paragraphs which follow.

Ground vehicles typically have four types of events:

1. firing while stationary ($\overline{M}F$),

2. moving and not firing ($M\overline{F}$),

3. neither moving nor firing ($\overline{M}\overline{F}$), and

4. moving and firing simultaneously (MF).

The procedure for determining the event time depends upon which of the four events transpires.

The calculation of the time for a firing without moving event, $\overline{M}F$, depends on the manner in which the weapon is fired. An event for a weapon that is fired by adjusting the aiming point after each round includes loading, laying, and flight times. A tank main gun or an antitank weapon would be fired in this manner. The event commences once the decision to fire is made and terminates when the projectile impacts. It should be noted that the loading activity may be occurring simultaneously with laying or projectile flight. The firing time t_f is determined through a Monte Carlo sampling from the pertinent

firing time distribution. For rapid-fire weapons, the firing occurs for a specified time interval t_{rf}; e.g., ten seconds. The number of rounds fired then becomes a variable dependent on the weapon's rate of fire.

The length of a movement event without firing activity; i.e., $M\overline{F}$, is defined by a fixed time interval and the distance traveled becomes a function of the movement time t_m. When obstacles or terrain seriously reduce an element's mobility, allowing a change in the direction of movement at the end of a time period gives the element more flexibility than forcing it to travel an arbitrarily prescribed distance.

Event times for the remaining cases are specified by applying two rules:

1. For an MF event, set the event time equivalent to the firing time t_f.

2. For an $\overline{MF}$ event, set the event time equivalent to the movement time t_m.

In DYNCOM, aerial vehicles are assumed to be moving at all times while they are over the battlefield. Thus, they have two types of events:

1. moving and not firing ($M\overline{F}$), and

2. moving and firing simultaneously (MF).

As explained in Chapters 6 and 7, the event times for aerial vehicles are found deterministically.

Typical forward observer events are target selection, data preparation, communication of a fire request, waiting for a fire mission to be executed, and communication of a fire adjustment. Target selection events are of constant duration and are repeated until the FO identifies a target. Data preparation and communication event times are determined by Monte Carlo sampling from pertinent time distributions. The length of time that an FO waits for a mission to be executed is variable and is determined by the activities of the firer that is to deliver the requested fire.

The fire-support coordinator element operating in the passive monitoring mode becomes active only when FO's request fire. In this case, the FSC has an event of zero time duration unless a message to the FO is required. Then, the event time is determined stochastically and represents the duration of the message. In the active mode, the FSC is again inactive until a message

is addressed to him. Then, the event time is once more the stochastically determined communication event time.

The ground-to-air communicator element has two types of events. One event is of fixed duration and occurs when the GAC is attempting to locate targets for the aerial teams. Such events are repeated until a target is located. When a target exists, a communication event transpires in which a request for fire is transmitted. This event time is determined stochastically and represents the duration of the message.

The radio nets have plot and stand-by events. During stand-by, the element is inactive, but when activated for a plot event, the event length is determined by Monte Carlo sampling from pertinent distributions. Artillery firing batteries are treated in a similar fashion. However, the active event is a firing event instead of a plot event.

The MISTIC launcher fire-support element and the aerial unit decision element also have events that are similar. Typical events are selection of a mission to be fired from those requested, communication with a forward observer or other fire-support element, and waiting for a response from a forward observer. The methods used to determine event times are similar to those used for forward observer events.

As stated previously, the processing sequence for the events outlined above is ordered in time by using "clocks" which are set for the time that each element will complete its present event. While the simulation is processing a given event, this event is called the "current event," and the element performing the event is called the "current element." Once a current element has been processed, its clock is updated by the current event time; then, the next current element to be processed can be determined by searching each clock to find the clock with lowest time. Figure 1.1 illustrates the selection of the next current event from a set of element clocks, and the beginning and end of each event are noted in the figure by vertical tick marks.

The entire battle is represented by a repetitive cycle of selecting a current element, determining its actions during the current event, and then selecting another current element. The battle is started by putting small random numbers in each clock, and then searching for the clock showing the minimum time. The battle is ended when one of the three conditions noted below is met:

1. The attacking force has seized all of its objectives,

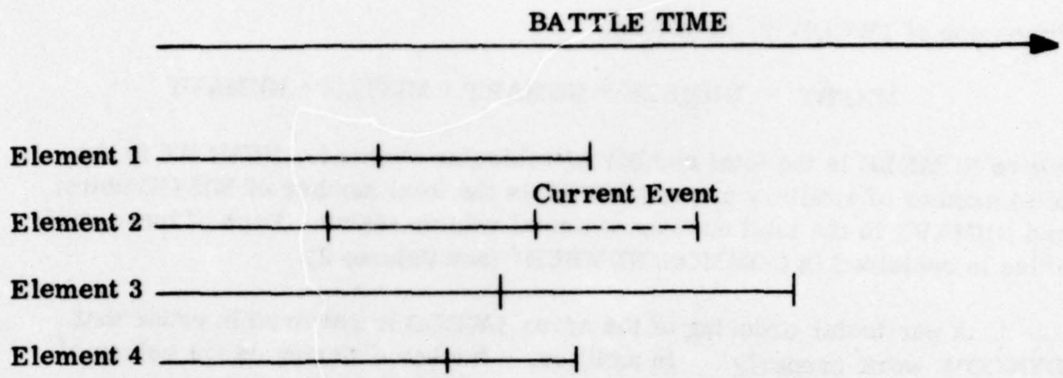


Figure 1.1.--Selection of the Next Current Event

2. The attack has been aborted, and the attacking force has been forced to assume defensive positions, or

3. One of the opposing forces has been annihilated.

Weapon Codes

Another important concept utilized in DYNCOM is that of a weapon code. For a vehicular element, a given weapon code implies that all elements having that code are identical with respect to combat characteristics. Such characteristics might include armament arrangements, degree of armor protection, communication gear, power plant, and so on. However, the concept has been expanded somewhat in the DYNCOM fire-support model. In this model, we refer to a fire-support weapon code. This code allows for a more detailed classification of supporting fire weapon units than is permitted by merely classifying a weapon unit according to whether it is an artillery, MISTIC, or aerial unit. The concept also permits a more convenient description of the characteristics of a particular supporting fire unit; it allows the grouping of multiple units having identical combat characteristics. For example, the fire-support weapon code for an artillery unit might imply the initial ammunition supply, the number of tubes in the battery, the accuracy of fires, the communications gear, and even the type of gun possessed by the battery. Similar characteristics might be cited for MISTIC units and aerial vehicle teams.

The weapon codes of all vehicular elements and all supporting fire units being simulated in DYNCOM are specified by the input array LWCOD. The

dimension of LWCOD is, therefore,

$$MNFRT = NUMELE + NUMART + MISTUN + NUMAVT$$

where NUMELE is the total number of vehicular elements, NUMART is the total number of artillery units, MISTUN is the total number of MISTIC units, and NUMAVT is the total number of aerial vehicle teams. Each of these variables is contained in COMMON/NUMBER/ (see Volume 2).

A particular ordering of the array LWCOD is required in order that DYNCOM work properly. In addition, a further ordering on the values of LWCOD for supporting fire units is assumed. LWCOD is a strictly monotonically increasing function for supporting fire units. That is,

$$LWCOD(I) < LWCOD(I + 1)$$

where I is the supporting fire element number and $I > NUMELE$. Table 1.3 summarizes the ordering required. The variables used there are defined in COMMON/NUMBER/, in Volume 2. Because the values are monotonically increasing, we can also conveniently subclassify the fire-support weapon codes into artillery unit weapon codes, MISTIC unit weapon codes and aerial unit weapon codes. This classification permits a more concise categorization of data that applies to only one or two of the fire-support unit types as opposed to all three types. For example, the MISTIC unit weapon code for MISTIC unit M is LWCOD(NT1) - MWART where

$$NT1 = NUMELE + NUMART + M.$$

Similarly, the aerial unit weapon code for aerial unit N is LWCOD(NT2)-MWMIS where

$$NT2 = NUMELE + NUMART + MISTUN + N.$$

Another observation to be made is that within TAPCOM II all aerial vehicles within an aerial section are identical even though the sections comprising a maneuver unit can be different. Thus, we arrive at the concept of an aerial section weapon code. This weapon code permits convenient classification of characteristics that pertain only to aerial vehicular elements. An aerial section weapon code is computed by the relation

$$LCOD = LWCOD(I) - MAXLWC$$

where I is the element number of any element (other than a fire-support element) in the aerial section of interest. The variable MAXLWC is defined as the maximum weapon code of nonaerial vehicular elements as contained in COMMON /NUMBER/.

Table 1.3

Weapon Code Ordering

Element Class	Array Order (I)	Value Order Required
(Ground) Vehicular (Aerial)	1 ⋮ NUMELE	1 ≤ LWCOD(I) ≤ MAXLWC LWCOD(I) > MAXLWC
Artillery Units (Blue)	NUMELE+1 ⋮ NUMELE+NBART	0 ≤ LWCOD(I) ≤ MWBART
Artillery Units (Red)	⋮ NUMELE+NUMART	MWBART ≤ LWCOD(I) ≤ MWART
MISTIC Units (Blue)	NUMELE+NUMART+1 ⋮ NUMELE+NUMART+NBMIS	MWART ≤ LWCOD(I) ≤ MWBMIS
MISTIC Units (Red)	⋮ NUMELE+NUMART+MISTUN	MWBMIS ≤ LWCOD(I) ≤ MWMIS
Aerial Units (Blue)	NUMELE+NUMART+MISTUN+1 ⋮ NUMELE+NUMART+MISTUN+NBAVT	MWMIS ≤ LWCOD(I) ≤ MWBAIR
Aerial Units (Red)	⋮ NUMELE+NUMART+MISTUN+NUMAVT	MWBAIR ≤ LWCOD(I) ≤ MWAIR

Aerial Vehicle Operations

Now that we have an overall description of the operation of DYNCOM and the integrated fire support model that has been developed we can discuss the operations of helicopters that are represented in DYNCOM. We will begin with a description of helicopter organizations and proceed with a description of the missions that these helicopters can perform.

Helicopter Organizations

Helicopters are basically vehicular elements. Therefore, they are organized into maneuver units for purposes of controlling their overall movement in the vicinity of the battlefield. The same team, platoon and section organizations outlined previously for other DYNCOM vehicular elements are used for helicopters.

A difference does exist, however, between helicopter maneuver units and other types of maneuver units. In ground maneuver units, each vehicle in the unit guides on the unit leader for the duration of the battle. Each element has a particular position that it should occupy with respect to the leader at all times. This position is determined by the formation patterns that are currently being used by the organizations within the maneuver unit. Helicopter maneuver units do not always operate as single entities, however. As will be discussed, it is possible for the sections of a maneuver unit to depart the maneuver unit formation and operate in a completely independent manner. Thus, the section is the basic organization used to control movement. Elements within a section are never allowed to operate independently of the section. Moreover, when casualties occur within a section, the section formation organization is revised to account for the missing element.

Helicopter Missions

From a previous discussion of the role of helicopters represented within DYNCOM, we recall that the helicopters provide both preplanned and immediate direct aerial fire support, and they also deliver countermeasure fires. We must now define more explicitly the missions of helicopters represented in DYNCOM.

Eight different missions exist for helicopters as shown in Table 1.4. The need for each of the missions is established by analysis of the requirements generated by other components of the fire-support model. We will discuss each of the missions in the paragraphs which follow. The variable IUNACT(NAT) shown in Table 1.4 is the activity state variable for aerial unit NAT and is used throughout TAPCOM II.

Table 1.4.--Helicopter Missions

IUNACT(NAT)	Mission	Mnemonic
1	Target of Opportunity	TOP
2	Counterbattery Attack	CBA
3	Search and Destroy	SAD
4	Self Defence	SDM
5	Indirect-Fire MISTIC	IFM
6	MISTIC Forward Observer	MFO
7	Counterbattery Observation	CBO
8	Special Forward Observer	SFO
9	Defensive Operations	DEF
10	Retiring from Battlefield	RET

The target-of-opportunity mission (TOP) is the mission flown by an aerial unit in response to a fire request from a forward observer (FO). The FO can be either airborne or on the ground, but in any case, the FO has requested fire from a specified unit and has transmitted his request over the ground-to-air radio net. The target consists of a known number of enemy weapons located within a short distance of the specified target complex location. However, any enemy element in the vicinity of the reported complex may be engaged with direct fire in a manner that will be described in more detail in a later paragraph.

Counterbattery attack missions (CBA) are also direct-fire missions. However, in this case, the target complex consists of the components of one specified enemy artillery battery. Again, the location of the target is specified but only the components of the battery may be attacked during a counterbattery attack mission. In order for other types of elements to be attacked, a self defense or countermeasure mission defined below must first be initiated. The CBA mission is requested by the artillery intelligence center element over the ground-to-air net.

The search-and-destroy mission (SAD) is another direct-fire attack mission. This mission is requested by the ground-to-air communicator element over the ground-to-air net. Again, the position of the target complex is specified in the target description. However, in this case, the composition of the target is unknown. In fact, it may be that there are no enemy elements

in the complex. The reason for this situation is that the SAD mission is always triggered by the movement of friendly ground elements. Prespecified data describes the location of possible enemy strong points and as friendly elements advance, the battlefield commander requests investigation of the various strong point locations by aerial teams. As will be discussed, any target element discovered at the specified strong point is eligible for attack with direct fire.

The self-defense mission (SDM) is more commonly referred to as a countermeasure mission. It is the only mission initiated by an aerial unit without request from some other fire-support element. The reason for this convention in TAPCOM II is that some mechanism is required to permit aerial units to act in their own defense or to interrupt existing missions in order to engage more lucrative targets. The SDM mission is conducted against a target of opportunity discovered by elements of the unit itself. The complex will consist of one or more enemy ground elements and any element in the area may be attacked with direct fire. The mission may be conducted by an aerial unit at any time a suitable target of opportunity is discovered.

The indirect-fire MISTIC mission (IFM) is one of the missions originally represented in TAPCOM and is included in TAPCOM II to provide a complete representation of MISTIC operations in DYNCOM. The possible need for augmentation of the MISTIC launcher force is constantly evaluated by the ground-to-air communicator element during a DYNCOM engagement. When it is determined that augmentation is desirable, an aerial unit capable of delivering indirect MISTIC fire is selected and is assigned to one of the MISTIC launcher units. The assignment is performed by the ground-to-air communicator over the ground-to-air radio net. Each section within the aerial unit is designated as a MISTIC launcher and for the duration of the mission, the various sections honor requests for MISTIC fire in a manner that is identical to other launchers. Upon each MISTIC firing assignment, one of the elements in the section performs the actual MISTIC launch operation. Since it is possible for aerial sections to become MISTIC launchers during an engagement, it is necessary that the input data set be constructed to allow for MISTIC launcher fire-support elements that are inactive at the start of the battle and which are not assigned to ground launcher vehicles. These fire-support elements can then be activated during the battle when they become associated with one of the aerial vehicles acting as a launcher vehicle. This topic is discussed in more detail in Chapter 6.

The MISTIC forward observer mission (MFO) is the other mission retained from TAPCOM. The ground-to-air communicator element constantly evaluates the need for augmentation of the MISTIC forward observer force during an engagement. When augmentation is required, an aerial unit consisting of elements equipped with MISTIC illuminators is selected and is assigned

to one of the MISTIC units. The assignment is transmitted over the ground-to-air radio net. Each section within the aerial unit is designated as a MISTIC forward observer and for the duration of the mission the various sections operate as any other MISTIC forward observer would. One of the vehicles in each section is designated as the transporting vehicle for the forward observer element.

Initiation and performance of the special forward observer mission (SFO) is similar to the MFO mission. However, in this case, elements of the aerial unit selected for the mission do not have to be equipped with illuminators. Moreover, the aerial unit is not assigned to any other fire-support unit. The sections operating as forward observers act as any other special forward observer would.

Now, since aerial sections may become MISTIC and special forward observers, it is necessary, as in the case of MISTIC launchers, to allocate space in the input data set for inactive and unassigned forward observer elements. There should be enough space to accomodate as many aerial sections as may possibly be simultaneously assigned as FO's. Also, the unassigned FO numbers should appear both in the special and the regular FO regions. See Chapter 6 for a more detailed discussion of this topic.

The final mission for aerial units is the counterbattery observation mission (CBO). This mission is requested by the artillery intelligence center element over the ground-to-air net and it is primarily an observation mission. A particular enemy artillery battery is specified as the target and its approximate location is conveyed to the aerial unit. The purpose of the mission is for elements of the aerial unit to attempt to detect the battery and localize it so that more accurate counterbattery fire may be delivered upon it. No firing of any type is allowed by the aerial elements conducting the mission.

In additon to the eight missions outlined above, there are actually three other operations permitted of aerial units that are not actually missions. When IUNACT(NAT) = 0, aerial unit NAT is over the battlefield but is not performing a mission. This activity is included to allow for times during the battle when there is no mission for an aerial unit. Such an activity is denoted by the mnemonic NTA.

Another activity is known as defensive operations and is indicated by IUNACT(NAT) = 9 (mnemonic: DEF). This activity occurs when the heat of battle is such that an aerial unit finds it advisable to operate in a defensive fashion. The unit aborts any previously assigned mission and seeks a safe position. For a period of time, the unit is unable to accept any other mission assignment.

The final activity state is denoted by IUNACT(NAT) = 10 and the mnemonic RET. This state exists at the start of a battle when aerial unit NAT has yet to commence its first mission. The state also may exist later if NAT has found that it can no longer operate as an effective unit. In the first case, it is assumed that NAT is not actually over the battlefield and consequently is not actually a participant. Elements of the unit are not permitted to accumulate intelligence nor are they allowed to engage targets. On the other hand, they cannot be detected or engaged by enemy weapons. In the latter case, NAT is said to be retiring. When this state exists, elements of the unit commence movement off the battlefield and when they reach the edge, they are removed as participants in the battle. While enroute, they are subject to attack but cannot return fire. Once removed, they are never returned to action.

Mission Phases

Each mission or activity performed by an aerial unit is divided into two phases and the activities permitted of the elements within the unit are dependent upon the mission phase. The phase indicator for aerial unit NAT is IPHASE(NAT).

To delineate between phases, the concept of a mission operations area is used. This area is a circle of radius specified by input data and centered at the objective position of the mission. The objective position is either the reported target location (for the missions: TOP, CBA, SAD, SDM, and CBO) or is some other position at which the mission activities are to be concentrated (for the missions: IFM, MFO, SFO, NTA, DEF, and RET). Rules for selecting these locations are discussed in Chapter 4.

When the unit is outside the mission operations area, the unit is said to be in the enroute phase of the mission and IPHASE(NAT) = 1. During this phase, all sections of the unit that are operating with the unit fly in the maneuver unit formation. Thus, during the enroute phase, an aerial unit moves in a fashion similar to that of ground units. Each element guides on the leader of the formation and the leader moves toward the mission area along a route that is selected dynamically to provide protection for the elements of the unit. The desired route is of constant altitude above the terrain (nap of the earth).

Upon entering the mission area, the phase indicator IPHASE(NAT) is set to zero. This indicates that mission area operations have commenced. Then, the sections of the unit are allowed the activities indicated by the entries in Table 1.5. The reader will note that these activities are consistent with the objective of each mission, and the operating rules are specified to be in agreement with guidelines established at meetings of the Study Advisory Group monitoring the research for the System Analysis Group. The reader will also note that the self-defense mission (SDM) or countermeasure mission has no enroute

Table 1.5

Mission Area Operating Rules

IUNACT(NAT)	Mnemonic	Activity
0	NTA	Sections of the unit operate in the unit formation while occupying a loiter station. No attacks of any type are permitted.
1	TOP	Sections of the unit choose independent routes to be used in searching for targets. Any target complex discovered by a section may be attacked with direct fire.
2	CBA	Same as TOP except the assigned enemy battery is the only target complex that may be attacked.
3	SAD	Same as TOP.
4	SDM	Same as TOP.
5	IFM	Sections of the unit operate in the unit formation while occupying a loiter station. Sections may leave the formation to honor requests for indirect MISTIC fire but return upon completion of the attack.
6	MFO	Operations are similar to TOP except that the sections operate as forward observers requesting fire support. No direct-fire attacks are allowed.
7	CBO	Sections of the unit operate in the unit formation while flying a route that permits search for the assigned artillery battery. No attacks of any type are permitted.
8	SFO	Same as MFO.
9	DEF	Same as NTA.
10	RET	Upon retirement, sections of the unit are removed from the battle. Therefore, no activity is represented.

phase. That is, it is assumed that when a decision is formulated to commence an SDM mission, the target of opportunity is in close proximity to the elements of the unit.

From the above descriptions, it is obvious that during some missions attacks are permitted while other missions permit only observation. During some missions, no activity other than movement is allowed. To specify this more general description, the mission class indicator MCLASS(NAT) is used throughout TAPCOM II for aerial unit NAT. The definitions are as follows:

$$
\text{MCLASS(NAT)} = \begin{cases} 1 & \text{indicates a movement only mission} \\ & \text{(NTA, DEF, RET),} \\ 2 & \text{indicates a direct-fire mission} \\ & \text{(TOP, CBA, SAD, SDM),} \\ 3 & \text{indicates an observation mission} \\ & \text{(MFO, CBO, SFO), and} \\ 4 & \text{indicates an indirect-fire mission} \\ & \text{(IFM).} \end{cases}
$$

Detailed Section Activity

The activities outlined above for sections are those that are allowed when a section is operating with its unit. However, it is possible for a section to operate independent of the unit, either temporarily or permanently. We will outline these situations in the paragraphs which follow. First, we must define the activity state indicators that are used for sections.

An aerial section NSEC has two activity state indicators that are similar in definition and use to those that are defined for units. The first is JUNACT(NSEC) and corresponds to IUNACT(NAT). The other is JPHASE(NSEC) and corresponds to IPHASE(NAT). The definitions of JUNACT(NSEC) are presented in Table 1.6.

The phase indicator JPHASE(NSEC) specifies whether or not section NSEC has arrived at the final position specified by the activity indicator JUNACT(NSEC). Again, a value of one indicates enroute movement and a zero is used otherwise. When JPHASE(NSEC) = 0 and JUNACT(NSEC) < 2, section NSEC is in its proper position with respect to other sections of the maneuver unit formation. When JPHASE(NSEC) = 1 and JUNACT(NSEC) < 2, section NSEC is in close proximity to the unit formation but is transitioning into the formation from some other independent movement trace.

A value for JUNACT(NSEC) of two indicates that an attack is underway. The attack can be with direct or indirect fire or it can simply be that section

Table 1.6

Section State Variable Definition

JUNACT(NSEC)	Activity
0	Unit is enroute to a mission area and section NSEC is flying with the unit formation.
1	Unit is in mission area and section NSEC is loitering or searching for targets as specified by the mission operating rules.
2	Unit is in mission area and section NSEC is either delivering direct or indirect fire or is illuminating targets as a MISTIC forward observer.
3	Unit is in mission area and section NSEC is waiting for fire support requested while acting as a forward observer.
4	Section NSEC is retiring from the battlefield independent of the unit.
5	Section NSEC is operating defensively independent of the unit.
6	Section NSEC is returning to the unit after having operated defensively independent of the unit.

NSEC is illuminating a target for indirect-fire MISTIC attack. Routes are constructed so that fire or illumination commences at a point called the initial point and until this point is reached JPHASE(NSEC) = 1. Thereafter, JPHASE(NSEC) = 0 for the duration of the attack. The situations in which direct and indirect fire and illumination occur are indicated in Table 1.5.

A value of three for JUNACT(NSEC) indicates that a section belonging to a unit performing a forward observer mission has located a target for fire support and is waiting for the fire to be delivered. When JPHASE(NSEC) = 1, the section is enroute to a waiting position. When the waiting position is achieved, JPHASE(NSEC) = 0 and section NSEC begins to loiter independent of the rest of the unit.

When a section becomes ineffective as a combatant organization, it may retire independent of the rest of the unit. Such a decision is made when significant casualties occur within the section, when ammunition is exhausted or when fuel supplies within the section become short. A section that begins to retire will never return to the battle and JUNACT(NSEC) = 4. While section NSEC is enroute off the battlefield, JPHASE(NSEC) = 1. When the retirement operation is complete, JPHASE(NSEC) is set to zero.

In the process of operating in a mission area, it is sometimes more advisable for a section to seek a defensive position than it is for the section to continue searching for a target or to commence an attack against a target. It may be that the enemy threat is too great to allow normal operations to continue. When this occurs, the section determines a neutral position that offers protection and occupies this position for a period of time sufficient for the situation to improve. Then, the section can rejoin the unit. Independent defensive operations for a section are indicated by JUNACT(NSEC) = 5.

While enroute to a defensive position, JPHASE(NSEC) = 1. Thereafter, as long as the section loiters at the defensive position, JPHASE(NSEC) = 0.

When a section decides to rejoin the unit after operating defensively, the section commences enroute movement. When this occurs, JUNACT(NSEC) is equal to six and JPHASE(NSEC) is equal to one. Upon reaching the vicinity of the unit operating area, JUNACT(NSEC) is reset to one and JPHASE(NSEC) remains one. This convention will cause the section to rejoin the unit formation. Thus, the combination JUNACT(NSEC) = 6 and JPHASE(NSEC) = 0 never arises.

Initializing Helicopter Operations

The model is designed under the assumption that at the start of a battle helicopter units are not over the battlefield and are not operating in any way against the enemy. In fact, they may not even be available for mission assignment for some period of time after the start of a battle. To implement these assumptions, the model performs according to the scheme outlined in the following paragraphs and uses data that have been initialized as indicated.

The state variables for all helicopter units must be initialized by the user to indicate whether or not they are available for a mission at the start of the battle. Three arrays of variables are involved. A value of ten is used for all entries in the array IUNACT. This value indicates that the units are not operating over the battlefield. Then, if unit NAT is available at the start of the battle, IPHASE(NAT) should be zero. This value indicates that the unit is available. If the unit is not available, IPHASE (NSEC) should have the value two. This value indicates that the unit is enroute to the battlefield at the start of the battle. The final variable is TIMARR(NAT). This variable indicates the battle time at which the unit will arrive in the vicinity of the battlefield and will become available.

The other variables that must be initialized by the user specify the position over the battlefield at which the leader of each maneuver unit will appear when the first mission for the unit is assigned. No helicopter element positions other than those for the aerial unit leaders must be initialized. The positions of the remaining elements in the units are computed by the model according to the unit formation specifications.

The model initializes the clock times of all helicopter elements to large positive values. These clock times are not revised until the first mission is assigned. Then, the clock times are set to the battle time at which the mission is assigned. In this way, helicopters are prevented from being processed as current elements until they have actually commenced mission operations.

In setting the clocks of the elements within an aerial maneuver unit, a scheme is used to insure that the maneuver unit leader will become the first element of the unit to be processed as the current element. The scheme also insures that the leader of each section is processed before any other member of the section. The scheme is used for several reasons. First, it is necessary that the maneuver unit leader be processed before any other element since a considerable amount of bookkeeping is required any time a new mission is assigned. This situation holds, of course, in the case of the initial mission. Next, TAPCOM II operates on the basis that a section leader makes all movement and firing decisions for the section. Other elements of the section simply

move and fire in response to these decisions. Therefore, the clock of a section leader is always maintained slightly in advance of other section elements so that the leader is always processed first.

DYNCOM

Now that we have discussed the various operations that are simulated by DYNCOM, we may describe the processing that occurs within the DYNCOM program. This program employs a modular structure. Not only does this structure facilitate understanding of the program organization but each module that is employed forms a concise processor for one of the types of elements that have been discussed. Although these modules represent separate identifiable parts of the program, their functioning is interrelated in that information is passed from one module to another. The models and corresponding program modules can be revised to represent different combat situations or element capabilities, and various combinations of these modules can be assembled as required. Other modules can be easily added to represent types of weapons not presently represented.

Armor Module

When a ground vehicle is the current element, the beginning of each event provides an opportunity for tactics to be reviewed and altered in response to changes in the battle situation occurring in previous events. Accordingly, the Armor Module is designed to evaluate the battle situation at the onset of each event before an element is committed to action in its current event.

The first step in processing a ground vehicle element is to transmit messages on the nets which it is monitoring to give it the benefit of intelligence provided by other friendly elements. After messages have been received, the intelligence acquired by visual search during the current element's previous event is determined. Also, intelligence lost by the current element due to loss of intervisibility or to its becoming neutralized is assessed. If the current element is a maneuver unit leader, it can evaluate its intelligence and the battle situation, and then change its unit's movement plan if required. Each element evaluates its firing activities prior to executing the current event. The final steps in the computation procedure determine the outcome of movement and firing activities performed by the current element in the current event.

To accomplish the above processing, a sequence of subroutines exists to perform each of the tasks described. The functions performed by each subroutine are described below:

Communications.--In order to assess communication time delays, the Communications Subroutine (subroutine COM, reference 14 and Chapter 5) represents communication traffic on platoon tactical nets, company tactical nets, and battalion tactical nets. Messages reporting newly detected enemy weapons are explicitly described. When messages are originated for transmission and the nets are busy, queues are formed so that these messages can be transmitted at a later time. In order to disseminate information throughout this tactical organization, the Communications Subroutine will originate messages on the battalion net after they have been sent first on a platoon net and then on a company net. In fact, messages are disseminated throughout the net structure regardless of where they originate.

Intelligence.--An intelligence list is maintained by the Intelligence Subroutine (subroutine INTELL, reference 14 and Chapter 5) for each element giving the enemy weapons that he has approximately located, visually detected, or pinpointed, and this list is updated each event as intelligence is gained or lost. Approximate knowledge is possessed when the enemy element was reported on a communication net or when the enemy element was previously detected but is no longer intervisible. A visual-detection model is used in determining detection times. This model considers variables which affect the detectability of a weapon such as range, concealment, crossing velocity, and scene complexity. Also, both the neutralization status of the observer and the firing activities of undetected enemy weapons are considered in applying this visual detection model.

Movement Controller.--The Movement Controller Subroutine (subroutime MVCON, reference 2 and Chapter 9) represents the selection of routes, formations, and desired unit speeds by maneuver unit leaders. The timing of withdrawal and advance movements by delaying and supporting fire units is specified by the Movement Controller. Extensive use is made of the route-selection and formation-selection submodels. Assault routes are computed in the vicinity of the axis of advance in order to determine routes that have desirable trafficability, cover, and fields of fire; and the commander's intelligence concerning enemy weapons, strong points, and terrain conditions is considered. Using the locations of detected enemy weapons, the principal direction of threat is identified in order to determine formations and desired speeds for mobile maneuver units. If the unit is in a minefield, the decision to breach the minefield, traverse the minefield, or perform a retrograde movement out of the minefield is made by the Movement Controller. The timing of withdrawal of units at outposts is also controlled by the Movement Controller.

Fire Controller.--As enemy elements are detected, the Fire Controller (subroutine FIRCON, reference 3 and Chapter 9) makes the decision to engage a target, determines the highest priority target, monitors the length of the fire

mission, determines whether the target is to be engaged while the firer is moving, directs the firer to a fire position, selects a projectile, and maintains the ammunition supply records. In selecting targets, the Fire Controller considers such factors as:

1. target range,
2. target weapon type,
3. target cover,
4. projectile effective ranges,
5. firer's sector of responsibility,
6. whether the target is firing,
7. whether the target is firing at the firer,
8. whether other friendly weapons are engaging the target, and
9. unit integrity.

Also, targets can be transferred to other firing units in order to represent fire and movement tactics. The reader should note that only direct-fire is considered in the Fire Controller, with indirect fire of MISTIC launcher vehicles being controlled in another module to be discussed. If a MISTIC launcher vehicle selects a direct-fire target, then the ammunition selected can be either a conventional round or a MISTIC missile depending upon input selection criteria and ammunition supplies.

Movement.--Using the unit formation and route designated by the Movement Controller, the Movement Model (subroutine MOV, reference 3) computes the current element's desired position at the end of his current event. This desired position is determined relative to the position and speed of the unit leader so that the unit maintains formation integrity. The element is placed at its desired position if it is capable of traveling the desired distance. Otherwise, the new element position is computed as the farthest point it can achieve along its route. The calculation of vehicle mobility capability considers the vehicle physical characteristics and the mobility environment along its movement path. If the vehicle enters a minefield, the Movement Model determines whether a mine is detonated and damage occurs.

Firing. --The type of round to be fired is specified by codes prepared by the Fire Controller when a target is selected. If the round is a conventional ballistic round, accuracy and lethality are assessed by the Firing Model (subroutine FIRMOD, reference 3 and Chapter 11 in reference 15) using the uncovered target profile, target range, projectile dispersion and target vulnerability characteristics. Given a hit, the target is placed in a neutralization status. If the round selected is one of the types of missiles represented in DYNCOM, then one of the missile operations models is used to implement the firing. If the missile is of the beam-rider type, the missile is flown to the target by a second order control model and lethality is assessed after it is determined whether or not a hit occurs (subroutine SHILLY, reference 7. If the round selected is a semiactive homing missile (MISTIC), a MISTIC missile element is created and control is relinquished to the MISTIC missile flight model (subroutines FLIGHT, BFLITE and FINAL , Chapter 2. Representation of the flight of such missiles requires a series of events for the missile element so lethality is not assessed by the firing model when the firer is current. Instead, lethality is determined when the missile impacts.

TAPCOM II

When an aerial vehicle element is current, a sequence of models similar to those used in the Armor Module are employed to represent the activities of the element. Again, the first step in the processing sequence is to transmit messages on the nets which the current element is monitoring. This processing is accomplished by the Communications Subroutine described previously. The next step is to revise the intelligence that the current element has. Again, the Intelligence Subroutine described previously is used. The next step in the processing sequence is the evaluation of intelligence and the battle situation in order to arrive at new movement and firing decisions. This processing is accomplished in subroutine HELCON (Chapter 6 which processes only aerial vehicle section leaders. The decisions that are made apply to each element in the section. The final steps in the computational procedure determine the outcome of movement (subroutine HELMOV, Chapter 7 and firing activities (subroutine HFIRE, Chapter 8) performed by the current element in the current event.

A deviation of the processing outlined above occurs when helicopter elements become casualties. This deviation is the result of the way in which casualties are assessed for air defense weapons. When an air defense weapon has a firing event and inflicts damage on a helicopter element, only the extent of damage inflicted is determined during the air defense weapon event (subroutine CASHEL, Chapter 9). Removal of a casualty helicopter from the battle is deferred until the casualty element becomes current. Thus, prior to updating communication for a helicopter current element, subroutine GETHEL (Chapter 9) is called and it is determined whether or not the element is a casualty. If so, the organization containing the casualty is adjusted. Then, if

the element is not the sole survivor of its section, its event is terminated and it is removed from the battle. If it is the sole survivor, it is allowed to be processed through the movement and fire controller and then it is removed (subroutine HRAPUP, Chapter 9). This latter procedure allows certain book-keeping that is required when an aerial section becomes inactive.

Missile Module

As previously discussed, semi-active homing missiles are treated as separate elements after they are launched. The Missile Module represents the flight and terminal effects of such elements (subroutines FLIGHT, BFLITE and FINAL, Chapter 2).

Fire Support Target Selection Module

There are three subroutines within the Fire Support Target Selection Module. The first (subroutine AFO, Chapter 2 in reference 15) treats the activities of forward observer elements. These elements select targets of opportunity for artillery, MISTIC and aerial units and may also request on-call artillery fires if required. Time delays for preparation of target data and for communication are assessed within this subroutine.

The second subroutine (subroutine AIC, reference 13 and Chapter 2 in reference 15) represents the activities of the artillery intelligence center element in controlling counterbattery fires. Requests for fire may be addressed to either artillery or aerial units, and in addition, aerial units may be called upon to perform counterbattery observation.

The final subroutine (subroutine AFSC, Chapter 2 in reference 15) represents the activities of the fire-support coordinator element and the ground-to-air communicator element. The FSC is processed when an FO requests assistance in selecting a fire support means. The FSC responds by selecting an appropriate artillery, MISTIC or aerial unit to deliver the requested fire. Communication time delays are assessed. The GAC continually monitors the battle to determine when a need exists for additional launchers and forward observers and when search-and-destroy missions should be performed by aerial units. Requests for each of the mission types are transmitted to selected aerial units when required and communication delays are assessed.

Fire Support Firer Module

Three subroutines are also contained within the Fire Support Firer Module. The first (subroutine AFB, reference 13 and Chapter 4 in reference 15 represents the activities of artillery firing batteries required in execution of

scheduled, on-call, target of opportunity and counterbattery missions. Time delays for delivery of fires are assessed and lethality against targets attacked with the fires is determined.

The next subroutine (subroutine MFB, Chapter 3 represents the communication activities of the indirect-fire MISTIC launcher fire-support element. All communications required in mission negotiations with a forward observer are simulated and corresponding time delays are assessed.

The final subroutine (subroutine AIRFB, Chapter 4 treats the activities of each aerial unit decision element. This element controls the overall movement activity of an aerial unit by formulating decisions to be implemented by the aerial elements that are processed through TAPCOM II. All communication activities required with other fire-support elements are simulated and time delays for these activities are assessed.

Radio Net Module

The Radio Net Module consists of one subroutine (subroutine AFDC, Chapter 3 in reference 15) to process the various radio net elements that are used in the DYNCOM fire-support model. When fire-request communications occur, a net becomes occupied with traffic and cannot be used again until the net clears. This occurs when the radio net element becomes the current element and is processed by the Radio Net Module. In the case of an initial fire request, delays for data processing are assessed. Also, processing occurs to interrupt communications over artillery radio nets when artillery units become casualties of counterbattery fire.

Summary

A schematic of the simulation program is shown in Figure 1.2, which presents the sequence in which the program subroutines are employed. The Sequence Controller updates the element clocks and determines the following current element. A flow chart of subroutine MAIN which performs the processing indicated in Figure 1.2 appears in Volume 2.

The design of many of the simulation models that have been added or revised in constructing DYNCOM is described in this volume. In Chapter 2, the flight of MISTIC missiles in either a ballistic trajectory of level flight mode is described. The operations of MISTIC launchers are described in Chapter 3, both with respect to direct-fire activities and indirect-fire activities. Then, in Chapter 4, the Aerial Unit Decision Element Model is described. Chapter 5

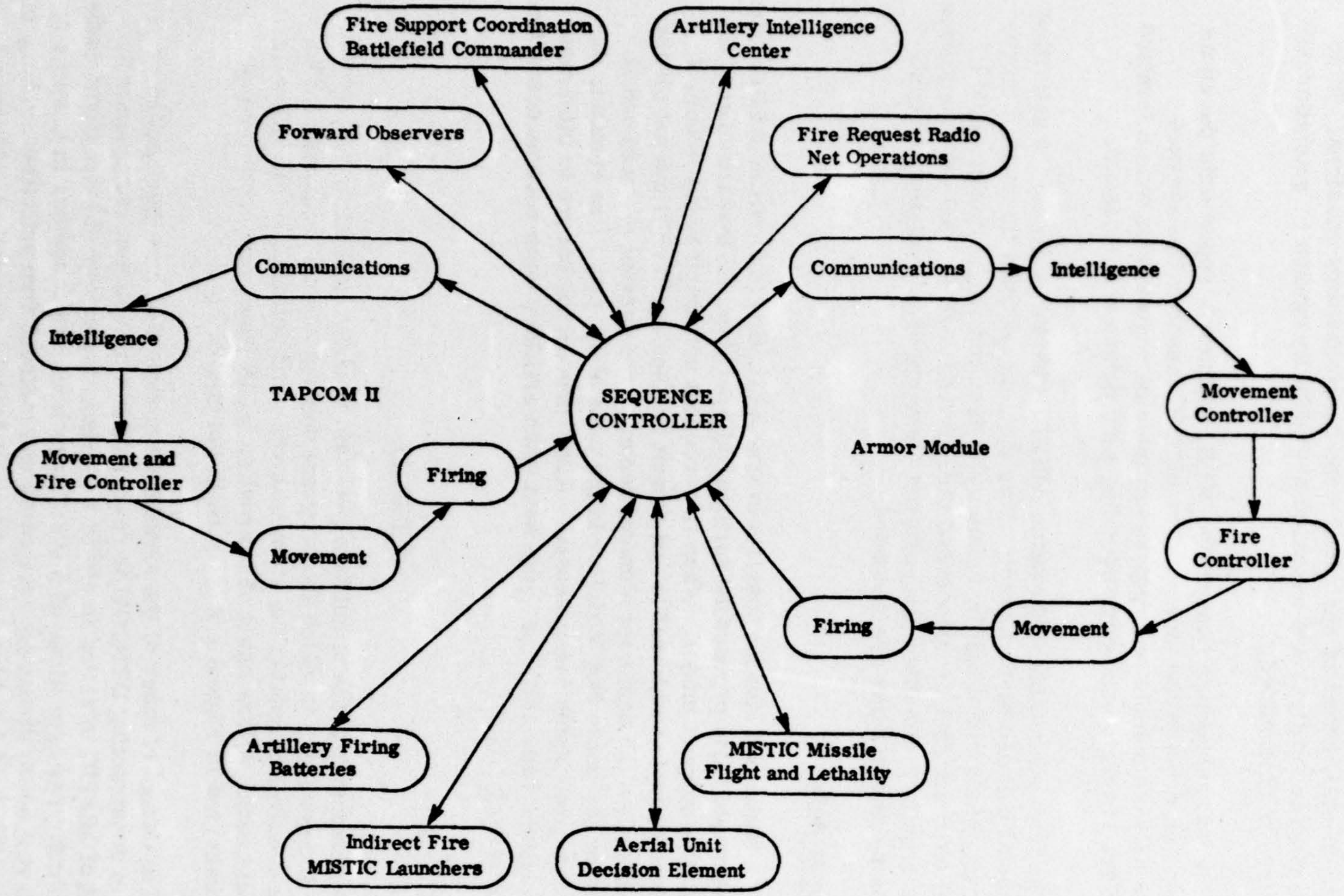


Figure 1. 2. -- DYNCOM Schematic

presents a summary of the Communications Model and the Intelligence Model and delineates revisions that have been made to improve the representation of aerial vehicle intelligence processes.

Chapters 6, 7, and 9 present descriptions of the new models that have been developed for TAPCOM II. In Chapter 6, the Movement and Fire Controller for aerial vehicle sections is described; the Aerial Movement Model is presented in Chapter 7; and Chapter 8 presents the Aerial Firing Model. Chapter 9 presents a collection of miscellaneous material that is necessitated by changes elsewhere in this volume.

VARIABLE DEFINITION INDEX

REFERENCES

1. Howland, D. and G. Clark, The Tank Weapon System, RF 573 AR 66-1 (U), Vol. 1, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1966, AD 801900.

2. Bishop, A. and G. Clark, The Tank Weapon System, RF 573 AR 69-2A (U), Vol. 2A, Systems Research Group, The Ohio State University, Columbus, Ohio, October 1969, AD 864919.

3. Bishop, A. and G. Clark, The Tank Weapon System, RF 573 AR 69-2B (U), Vol. 2B, Systems Research Group, The Ohio State University, Columbus, Ohio, September 1969, AD 864920.

4. Clark, G. and L. Moss, The Tank Weapon System, RF 573 AR 69-3A (U), Vol. 3A, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 864921.

5. Clark, G. and L. Moss, The Tank Weapon System, RF 573 AR 69-3B (U), Vol. 3B, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 864922.

6. Bishop, A. and G. Clark, The Tank Weapon System, RF 573 AR 69-4 (U), Vol. 4, Systems Research Group, The Ohio State University, Columbus, Ohio, September 1969, AD 864923.

7. Bishop, A. and G. Clark, The Land Combat Model (DYNCOM), RF 2376 FR-1 (U) Vol. 1, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

8. Clark, G., The Land Combat Model (DYNCOM), RF 2376 FR-2 (U), Vol. 2, Systems Research Group, The Ohio State University, Columbus, Ohio, January 1970, AD B001641L.

9. Clark, G. and D. Hutcherson, The Land Combat Model, The Aerial Platform Combat Operations Module, RF 2376 FR 71-3 (U), Vol. 3, Systems Research Group, The Ohio State University, Columbus, Ohio, May 1971, AD 887264.

10. Clark, G., et al., The Land Combat Model, DYNCOM Programmer's Manual, RF 2376 FR 70-4A (U), Vol. 4A, Systems Research Group, The Ohio State University, Columbus, Ohio, April 1971, AD 872508.

11. Clark, G., et al., The Land Combat Model, DYNCOM Programmer's Manual, RF 2376 FR 70-4B (U), Vol. 4B, Systems Research Group, The Ohio State University, Columbus, Ohio, April 1971, AD 872508.

12. Clark, G., et al., The Land Combat Model, Classified Annex, RF 2376 FR 71-5 (S), Vol. 5, Systems Research Group, The Ohio State University, Columbus, Ohio, January 1971, AD 517621.

13. Clark, G. M. and S. H. Parry, Small Unit Combat Simulation (DYNTACS X), Counterbattery Fire Models, RF 2978 FR 70-1 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1970, AD 882877.

14. Clark, G., et al., Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L

15. Clark, G. and D. Hutcherson, Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

16. Headquarters, Department of the Army, Attack Helicopter Gunnery, FM 1-40, March 1969.

CHAPTER 2

LEVEL AND BALLISTIC MISSILE FLIGHT

by
William R. Verry
Gordon M. Clark

Introduction

This chapter presents the model design for representing in DYNCOM an Indirect Fire Semi-Active Missile that searches for a target using either a level or ballistic flight trajectory. The original models, presented in reference 1, were developed for a missile using a level flight search trajectory. Computations are described in this chapter for extending the original model to incorporate a ballistic flight concept. Flow charts and common area descriptions are contained in Volume 2. The reader is referred to Chapter 3 and reference 1 for an overview of the Indirect Fire Semi-Active Missile or MISTIC Model since only the search trajectory modifications are described in this chapter.

The incorporation of ballistic flight required modification to both the flight mode and the target detection mode of the initial model. Level flight was simulated in the original model as the movement of a sensor at a constant altitude. The sensor was the only missile component that needed to be simulated in level flight because the missile was assumed to fly out of the battle area unless target detection occurred. Only when a target was detected was the missile considered to change course toward the target and thus be assessed as a weapon. In ballistic flight the missile is fired toward the target and may impact in the vicinity of the target. In addition, the target seeker of a level flight missile was simulated by considering the area scanned on the ground during a time interval as a rectangle. A missile flying a ballistic trajectory is constantly changing altitude. Thus, the rectangular scan pattern would be unrealistic for a ballistic trajectory missile. For this case a trapezoid is used to represent the area scanned on the ground during a specified time interval. Therefore, the addition of ballistic missiles requires modification of the search model, and requires that the terminal effects of ballistic missiles which do not acquire a target still be assessed upon impact. These modifications have been accomplished.

The addition of ballistic MISTIC missiles to DYNCOM was accompanied by a new comprehensive request procedure for supporting fire reported in other

chapters of this volume (see especially Chapter 3 and reference 2). To facilitate integration of ballistic flight with these improvements the missile location procedures and the terrain search procedures to locate targets in the field of view were removed from subroutine FLIGHT (reference 1) and put in individual subroutines. The procedures to locate a missile on the battlefield, and those to initiate a ballistic launch, are now in subroutine BFLITE. These procedures are described later in this chapter. The terrain search procedures to locate targets in the field of view have been placed in subroutine SEEKER, and are reported next in this chapter. The original missile flight models had subroutines to select new targets and forward observers during a flight. The new fire support models, described in Chapter 1, incorporated some of these functions and modified others. During indirect fire with the new fire support models, a missile is launched toward a target complex. When FLIGHT begins the missile flight for indirect fire, FLIGHT needs to search for a target. Therefore, a new subroutine to find targets for FLIGHT that is consistent with the new fire support models has been developed and is included in the last portion of this chapter as subroutine NUTARG.

The refinement of missile search and flight routines in DYNCOM has been accomplished in compliance with three basic design criteria which require that each component of the simulation 1) simulate combat performance, 2) relate design variables to the performance, and 3) accomplish these objectives with maximum simplicity. The model of ballistic flight developed for DYNCOM accomplishes these objectives by modifying the search phenomena to include trapezoid ground forms for the area scanned and using a ballistic trajectory to represent the flight path during the search.

The Trapezoid Model

The basic purpose of the trapezoid model is to identify the time that a target enters and/or leaves the missile field of view. This objective could be achieved with considerable computational expense by representing the projection of this field of view on the terrain profile at an instant in time, e.g., during a time interval of .1 second, and then checking for the presence of the target within this field of view. Of course, this computation would have to be repeated for each small time interval, e.g., .1 seconds. A more macroscopic representation was constructed for DYNCOM using the trapezoid model that has adequate resolution.

The trapezoid representation of the area scanned on the terrain profile is applied to a comparatively long time interval Δt, e.g., $\Delta t \leq 2$ seconds. The underlying concept for this trapezoid model is described using the basic assumptions of the level flight model. Assume that the missile is flying in a fixed direction at a constant altitude. Also, the missile has a sensor which rapidly scans from side to side or laterally in a direction perpendicular to the missile flight path. A view from the rear of the missile is shown in Figure 2.1, and the

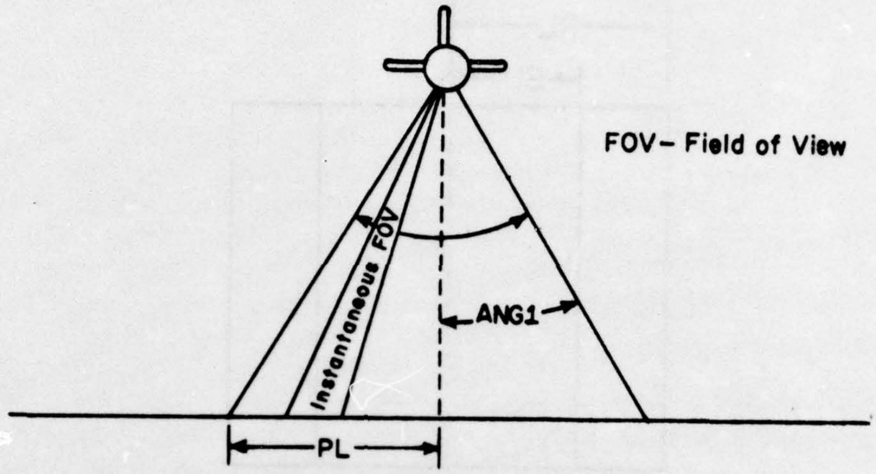


Figure 2.1.--Missile Scan Angle from Rear View

instantaneous sensor field of view is depicted. The outer limits (normal to the missile flight path) of this instantaneous field of view is defined by the angle ANG1 shown in the figure.

During a finite time interval, the area scanned by the missile in level flight is approximated by a rectangle as shown in Figure 2.2. The rectangular area within a distance of SL from the primary axis (dashed line) of the missile flight path is called the primary sensing area. The strip having a width of PL-SL is called the fringe sensing area. This fringe area has a lower target detection probability than the primary area. The purpose of the fringe area is to represent the fact that the lateral sensor scanning pattern will provide less coverage in the fringes of the sensing area. The distance PL is determined by projecting the angle ANG1 shown in Figure 2.1.

Note that a missile with a sensor having a wide field of view that does not scan laterally can be represented by this model. All that is required is for the fringe and primary sensing areas to be appropriately adjusted.

Also the assumption is made in representing areas scanned during a time interval Δt that the terrain is flat with respect to the missile. Of course, the missile altitude above this flat terrain profile can be constantly changing. Also, the terrain is certainly not considered flat with respect to weapon intervisibility relationships (forward observer and target).

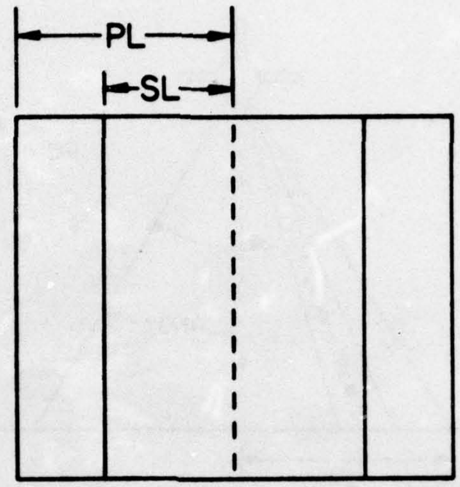


Figure 2.2.--Rectangular Area Scanned in Level Flight

For a ballistic search trajectory, the missile is either rising or falling. This rectangular area scanned is generalized to become a trapezoid as shown in Figure 2.3 for a rising missile. Note that the parallel sides of the trapezoid are normal to the horizontal projection of the missile flight path. Also, a falling missile would have a narrower sensing area in the direction of flight.

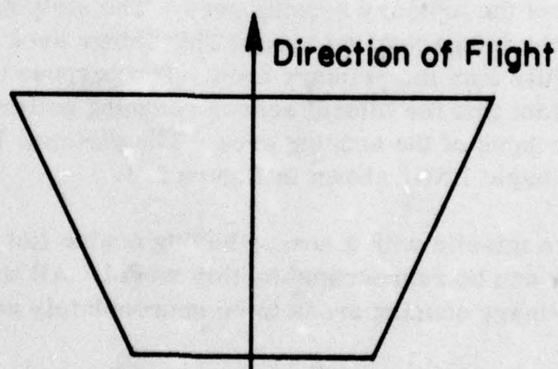


Figure 2.3.--Trapezoidal Area Scanned by a Rising Missile

Entering and Leaving the Field of View

During a specified time interval, the sensor system scans an area of terrain for a signal. If the target is not within this area, the missile cannot acquire the target, i.e., if the target is not within the terrain trapezoid, acquisition cannot occur. When search by the missile is initiated, SEEKER determines whether the target is within the missile's terrain trapezoid. If so, acquisition may occur; otherwise, SEEKER determines the point in time when the target just enters the missile field of view. That is, the time when the target is on the edge of the missile's terrain trapezoid. Similarly, SEEKER determines the time when the target leaves the missile field of view, unless acquisition occurs earlier. By the approach outlined above, a more efficient computational procedure has been developed than if FLIGHT repetitively checked to see if the target is within the missile field of view during a small time interval.

To implement the above procedure, the missile flight path must first be defined and then a procedure must be developed for projecting the search area on the terrain for a segment of this flight path. Of course, this search area has the form of a terrain trapezoid.

Determination of Ballistic Flight Path

The flight phenomena of a nonguided ballistic projectile in the atmosphere has been the subject of numerous research efforts. Laboratories have been investigating the relationship of propelling force or thrust to the other flight characteristics. The exterior ballistics of free flight as functions of shape and aerodynamic drag interactions has been extensively examined (see, for example, reference 2). When guidance is added, the relationship of missile design and operational characteristics has become a discipline of study (see reference 3). Numerous summaries of the interaction of free flight, guidance and aerodynamic factors have been prepared. For the development presented here, it is assumed that missile guidance will compensate for drag forces, and use of a simple parabolic flight path is justified. The flight calculations have been put in a separate subroutine, so that as new guidance systems and missile configurations are proposed for missiles to be simulated by DYNCOM, the subroutine can be changed and different flight paths can be substituted for the parabolic approximation.

The ballistic flight path for DYNCOM is simulated by a simple parabolic path using the initial velocity of the missile VM1 and the launch angle ANGLCH (see reference 4 as one source of the ballistic flight expressions). The down range distance RM1 between the launcher and the current missile location projected into the horizontal battlefield plane can be determined as a function of the flight time, T. The velocity in the horizontal plane is VM1 cos ANGLCH, and the down range distance is thus:

$$RM1 = T \cdot VM1 \cos ANGLCH.$$

The height of the missile above the terrain, ZM1, which corresponds to distance RM1 can be computed from the vertical component of the velocity less the loss due to gravitational pull. When g is the acceleration due to gravity, then the expression for ZM1 becomes

$$ZM1 = T \cdot VM1 \sin ANGLCH - gT^2/2.$$

The computations in DYNCOM require that periodically the distance RM1 and height ZM1 be computed for a specified T. A subroutine BFLITE has been developed to provide this information.

The introduction of ballistic flight into the simulation creates a need for a method in DYNCOM to compute a launch angle for each ballistic flight. This function is usually accomplished by the launcher crew before launch. However, the intent of subroutine BFLITE is to perform all ballistic computations in one subroutine to facilitate changes if other than parabolic flight paths are to be simulated. Thus, LNBFLT, a separate entry in BFLITE, has been developed to compute the launch angle for launch crews. These two subroutines were combined into one to enable users to change only one routine if a revised set of ballistic equations are to be used.

The launch angle is computed in LNBFLT, and computational equations are derived by solving the above equations for the missile flight point that is equivalent to the target position. The distance to the target from the launcher is stored in RMO. The launch angle is then computed from

$$ANGLCH = (1/2) \sin^{-1} (g \cdot RMO/VM1^2).$$

The equation is valid only for launch angles less than the angle for maximum flight range, 45°. Thus, it is necessary to check as to whether the missile traveling at speed VM1, can reach the target. Letting ANGLCH be 45° in the above equation, then the maximum flight range is computed as $RMX = VM1^2/g$. When a launch will require a range greater than RMX, the missile is launched at 45°, and guidance after target acquisition is assumed to modify the course and provide the extra range.

Computation of Flight Path

The development of the procedures for determining distance and height of missiles at a time T after launch has proceeded assuming that both level flight and ballistic missiles should be simulated by subroutine BFLITE. An array TYPMIS is used to identify the missile flight mode, and this array is in common TYPMIS. TYPMIS contains the missile type for each launcher, and

is indexed on the launcher number ILNNUM. Values in TYPMIS (ILNNUM) are given codes, C, which are coded as:

$$\text{TYPMIS(ILNNUM)} = \begin{cases} C < 3 \text{ for a level flight} \\ C \geq 3 \text{ for a ballistic flight} \\ C \text{ is odd for a passive missile sensor} \\ C \text{ is even for an E-O sensor system} \\ C < 5 \text{ for a sensor fixed to the missile} \\ C \geq 5 \text{ for a sensor oriented to the terrain} \end{cases}$$

(This coding is used in orienting the ballistic sensor, and its use is explained later in this chapter.)

The initial test in BFLITE uses this array, TYPMIS, to determine whether the flight profile is level or follows a ballistic trajectory. Other uses of TYPMIS will be described as they occur.

The flight of a missile is represented in DYNCOM by subroutine FLIGHT (reference 1) which uses BFLITE to locate the missile. Subroutine FLIGHT requests the range and height at time T, which is stored in common LNSET.

Level flight was simulated previously as a mean flight path with a random variation about the mean height (reference 1). This procedure has been retained in the current model when TYPMIS(ILNNUM) < 3. Level missile flight parameters are input in a set of arrays which represent the missile level flight curves as a function of down-range distance. Down-range distance is input in array RM(J) where the subscript J specified the particular distance point represented.[1] Altitudes are input in ZM(J) for mean altitude and ZMD(J) contains the standard deviation to be used to modify ZM(J). An initial Monte Carlo sample selects a basic deviate, Z1, from a normal random distribution with mean zero and variance 1; i.e., N(0,1). This deviate Z1 times ZMD(J) is used to modify ZM(J). For example, the altitude at down-range distance RM(2) would be ZM(2) + Z1 · ZMD(2). Similarly, the nominal or mean flight time to reach distance point RM(J) is TF(J), and the actual flight time is assumed to be normally distributed with standard deviation TD(J) at distance point J. The TD(J) values are recorded in common MD, and another normal deviate, i.e., Z2 is is used to modify TF(J). In some situations an interpolation between two points in time is required for time periods other than the intervals of length TF(J) + Z2 · TD(J) - TF(J-1) - Z2 · TD(J-1). The fraction of the time interval to be assessed is indicated in FLIGHT by a variable B1, which is also in

[1]Most common areas have the same names as the array names specified in this chapter; thus, the corresponding common name will not be mentioned.

common LNSET. When the time T to be assessed equals TF(J) + Z2 · TD(J), then B1 is set to one, and no interpolation is needed. However, when the time T to be assessed is less than TF(J) + Z2 · TD(J), it is necessary to compute the fractional increment in altitude and down-range distance with respect to ZM(J-1) and RM(J-1), respectively.

The flight model in FLIGHT also searched the terrain to find if the target was in the field of view, and when it was, if detection occurred. Each time the routine found an entry into the field of view, or sampled for detectiion, interpolation was needed to determine when the target entered the field of view. The recursive nature of this model has been retained, and the fraction of the time interval B1 between TF(J) + Z2 · TD(J) and TF(J-1) + Z2 · TD(J-1) is used in these routines to provide recursive interpolations.

The ballistic flight computations use the missile velocity input in array VM(ILNNUM) for the launcher processing the missile. The appropriate velocity for launcher ILNNUM is used for VM1. Ranges and heights computed during previous flight segments are stored in RM2 and ZM2, respectively. The distance from launcher to missile is computed as above from RM1 = T · (VM1) · cos(ANGLCH). The height likewise from ZM1 = T · sin ANGLCH - $gT^2/2$ + ZLP, where ZLP is the altitude of the launch point in common LNSET.

Finally, in ballistic flight the angle of the missile to the terrain continually changes as the missile flies the search path. To orient a guidance seeker fixed to the missile the angle of the missile to the terrain ANGMIS is necessary. The missile will be parallel with the terrain at one half of its ultimate range and have the opposite angle of the launch angle upon impact. The above equation for ZM1 can be restated as a function of RM1 as

$$\mathrm{ZM1} = \frac{\mathrm{RM1 \cdot VM1 \cdot \sin(ANGLCH)}}{\mathrm{VM1 \cdot \cos(ANGLCH)}} - \frac{\mathrm{RM1}^2 \cdot g}{\mathrm{2VM1}^2 \cos^2\mathrm{(ANGLCH)}}$$

$$= \mathrm{RM1\ \tan(ANGLCH)} - g\mathrm{RM1}^2/(2\mathrm{VM1}^2 \cos^2\mathrm{ANGLCH}).$$

Differentiating this expression with respect to RM1 will provide the slope of the missile heading, or the tangent of ANGMIS. Thus,

$$\frac{d(\mathrm{ZM1})}{d(\mathrm{RM1})} = \tan(\mathrm{ANGLCH}) - 2g\mathrm{RM1}/(2\mathrm{VM1}^2 \cos^2(\mathrm{ANGLCH})$$

or

$$\mathrm{ANGMIS} = \tan^{-1}(\tan \mathrm{ANGLCH} - g\mathrm{RM1}/(\mathrm{VM1}^2 \cos^2 \mathrm{ANGLCH})\,).$$

To compute the current missile angle the procedure finds the current missile height from the first expression for ZM1 above and then computes ANGMIS. The computed value of ANGMIS is used to orient sensors fixed to ballistic missiles.

The final step in computing the flight variables by BLFITE is to change the down-range distance, RM1, into map coordinates corresponding to the battlefield corrdinate system. The computing of the missile coordinates as XM, YM and placing them in common LNSET completes the calculations. For this calculation, the angle of the missile flight path to the major x-axis, ANGM in common OPEN, is used to compute the missile location XM, YM.

Computational Procedures

The procedures to establish the flight variables at time T by BFLITE are summarized below. Note that T, B1, Z1, and J (for a level-flight missile) are determined by FLIGHT and/or SEEKER.

1. If missile is flying a level course, if TYPMIS(ILNNUM) < 3, go to step 8.

2. Set VM1 = VM(ILNNUM) the current missile velocity for a missile launched by launcher ILNNUM.

3. Find the sine and cosine of the launch angle, ANGLCH.

4. Compute down range distance for ballistic flight
$RM1 = T \cdot VM1 \cdot \cos(ANGLCH)$.

5. Compute missile altitude;
$ZM1 = T \cdot VM1 \cdot \sin(ANGLCH) - 4.9 \cdot T^2 + ZLP$.

6. Compute the angle to the terrain
$ANGMIS = \tan^{-1} [\tan ANGLCH - 9.8\ RM1/(VM1^2 \cdot \cos^2 ANGLCH)]$.

7. Go to step 11.

8. Compute down-range distance for level flight path at time T,
$RM1 = B1 [RM(J) - RM(J-1)] + RM(J-1)$.

9. Determine previous missile height,
$B3 = ZM(J-1) + ZMD(J-1) * Z1$.

10. Determine current missile position
$ZM1 = B1 [ZM(J) + ZMD(J) * Z1 - B3] + B3 + ZLP$.

11. Compute map coordinates:

$$XM = XLP + RM1 \cos ANGM$$

$$YM = YLP + RM1 \sin ANGM.$$

12. The computations have been completed.

Computation of Launch Angle

The computation of the launch angle ANGLCH is accomplished in subroutine LNBFLT. This routine sets up variables for a ballistic flight. Initialization by this routine places the velocity for the missile in VM1 from the input velocity for each launcher, VM(ILNNUM). Next the range RM1 from launcher to target is determined using function RGXY(XLP, YLP, XAT, YAT) where XLP and YLP in common LNSET are the launcher coordinates and XAT and YAT are the apparent target coordinates described above. Finally, if the missile is to fly a level flight, TYPMIS < 3, then ANGLCH is not computed and processing is complete.

The computational procedure for the launch angle first determines if the target is within the range of a ballistic trajectory. The maximum range of the missile, RMX, is computed from $RMX = VM1^2/g$. If RMX is greater than the range to the target RMO, then a normal flight can be flown. For this flight the launch angle is

$$ANGLCH = (1/2) \arcsin (g \cdot RMO/VM1^2),$$

and the projected time of impact of the flight, TIMP in common LNSET, can be found from the launch time TM, also in common LNSET, by

$$TIMP = TM + 2 (VM1 \sin ANGLCH) / g.$$

For a flight which will be longer than the maximum range, the aim point must be designated as the furtherest point the missile can reach flying a ballistic trajectory. In this case the launch angle will be 45° and the apparent target center XAT, YAT must be set to the maximum distance for the missile. When the missile will fly its maximum distance the impact time becomes

$$TIMP = TM + \sqrt{2} \cdot VM1/g.$$

Level missile flight does not require a launch angle, but the time to fly past the target must be estimated. The level flight missile will fly at a velocity of VM1 for a range of RMO, thus the flight time is RMO/VM1, and fly-by time is estimated by

$$TIMP = TM + RMO/VM1.$$

Computations of Launch Angle

The computations of the initial conditions are accomplished in entry LNBELT to routine BFLITE. A summary of the procedures is listed below.

1. Set VM1 = VM(ILNNUM), the velocity of the current missile to the velocity appropriate for launcher ILNNUM.

2. Find the distance between the launcher and target, RMO = RGXY(XLP, YLP, XAT, YAT).

3. If missile is flying a level course, i.e., if TYPMIS < 3, go to step 13.

4. Compute maximum range $RMX = VM1^2/g$.

5. If within range, go to step 10.

6. Missile cannot reach target, compute impact point and set launch angle to 45°.

7. Locate maximum distance in map coordinates:
   $$XAT = XLP + (RMX/RMO)\ (XAT - XLP)$$
   $$YAT = YLP + (RMX/RMO)\ (YAT - YLP)$$

8. Compute impact time for a short flight
   $$TIMP = TM + \sqrt{2}\ VM1/g.$$

9. The computations are complete.

10. The missile can reach target, compute launch angle
    $$ANGLCH = .5 * \arcsin\ (gRMO/VM1^2).$$

11. Compute impact time for normal flight
    $$TIMP = TM + 2VM1 \sin\ (ANGLCH)/g.$$

12. The computations are complete.

13. Compute passover time for level flight
    $$TIMP = TM + RMO/VM1.$$

14. The computations are complete.

Computation of the Terrain Trapezoid

Relationship of Search to Flight

Following launch the flight of missiles is controlled until fly-by or impact by the previously designed subroutine FLIGHT (reference 1). The original FLIGHT routine contained a procedure which seeked a guidance return from an illuminated target for a rectangular search pattern from a level flying missile. The addition of a ballistic flight option requires addition of a trapezoid search pattern. To accomplish this the original rectangular seeking procedures were removed from FLIGHT and combined with the new trapezoid model in a new subroutine SEEKER. The resultant FLIGHT procedures process the flight of the missile up to the point where it is necessary to evaluate whether the guidance seeker system has a target in its field of view. At that point subroutine SEEKER is called and the value IGUIDE in SEEKER's calling parameters indicate to FLIGHT the result of the sensor search as follows:

IGUIDE =
- 0 not in field of view, continue search
- 1 missile flew past target, stop
- 2 the target is in the field of view

The search for a target in SEEKER consists of two basic sections, one for level flight and one for ballistic. The level flight section returns in IGUIDE a zero to indicate not in the field of view yet but may occur next time, a one to indicate not in field of view and cannot occur because the missile has flown past the target, and a two to indicate target in the field of view. The ballistic flight section returns either a two to indicate the target is in the field of view or a zero to indicate it is not yet in the field of view. Subroutine FLIGHT continues to fly the missile and returns to SEEKER as appropriate until the missile flys by; impacts, or the target is in the field of view. When the level flight return indicated that the missile has flown by the target the flight is terminated. When a ballistic flight does not find the target in the field of view and the altitude of the missile equals that of the terrain, then FLIGHT assesses the impact in the same manner used to determine the terminal effects of an artillery round.

The existence of a target in the field of view prompts FLIGHT to transfer control to subroutine FINAL (subroutine FINALE for an E-O missile) as previously described in reference 1. The target being in the field of view produces the chance of lock on. The model has been designed so that a level flying missile will fly off the battlefield and not be assessed unless the guidance system locks on a target. However, once the guidance is locked on a target, the level flying missile changes orientation and flys toward the target. Thus, once lock on occurs, in both the level and ballistic cases the missile will impact somewhere in the vicinity of the target, even if it loses guidance during the terminal phase.

The development of SEEKER and BFLITE required that flight location and search functions in FLIGHT had to be changed to calls to subroutines SEEKER and BFLITE. This resulted in changes to FLIGHT and some changes to FINAL and FINALE. In addition, the representation of the impact of a missile was accomplished by placing artillery assessment logic at the end of FLIGHT. These changes did not modify the basic logic as reported in reference 1 (and reference 5 for the artillery logic); therefore, a new comprehensive discussion is not included. However, enough changes in procedures were made to indicate that new flow charts would be helpful. The new flow charts of FLIGHT, FINAL, and FINALE appear in Volume 2 of this report.

Initialization in SEEKER

The search procedures in SEEKER begin by determining the location of the target, missile and launcher relative to one another. Subroutine FLIGHT provides the target element number ITARG to SEEKER through common LNSET. A call to ELOC provides the target location at its last event time TT. The missile is now at time T have been launched at time TM. Calls first to SPOT1 and then SPOT2 provide the current target location in (XT, YT, ZT). From these coordinates the horizontal range from launcher (at point XLN, YLN from common LNSET) to the target can be found using function RGXY.

The search process uses a new coordinate system which shifts from the battlefield coordinates to a new set which assumes the x-axis is pointed in the direction of flight and the origin is located at the launcher position. This increases initialization effort, but simplifies later field of view checks. To locate the target, the angle between the new x-axis and the target, THETA, is computed. Then new target coordinates (XP, YP) are found using THETA and the range between the target and the launcher. To simplify subsequent calculations, AYP is set equal to the absolute value of YP. These new target coordinates express the distance along the flight path from launcher to target as XP and the distance of the target from the missile flight path as YP. The next initialization step is the determination of the difference in altitude between missile and target, DELZ, from their altitude, ZM1 and ZT, respectively.

The type of missile and flight path must now be discerned and the appropriate procedures selected. Level flight produces a rectangular search pattern, ballistic flight produces a trapezoid. MISTIC sensors guide on the reflected impulse, E-O missiles use a different technique. Further, for ballistic flight the sensor may be fixed to the missile, changing its angle to the terrain, or it may be at a fixed orientation, always at the same angle to the terrain.

Each of these possibilities are identified through coding in array TYPMIS(ILNNUM). The type of flight was described above as TYPMIS < 3 indicates level flight, TYPMIS $\geq$ 3 indicates ballistic flight. An odd value of

TYPMIS indicates a MISTIC type sensor and an even value of TYPMIS an E-O sensor. One other coding is added for sensor orientation. If TYPMIS ≥ 5 the sensor remains at a fixed orientation to the initial launch angle. If TYPMIS = 3 or 4 then the sensor is fixed to the missile and its orientation with the terrain changes with the missile angle. Therefore, for each type of missile flight, sensor type, and orientation a different set of values will be initialized.

For ballistic flight, new values of the angles to the forward and rear edges of the field of view will be calculated for sensors fixed to the missile, since the angle changes between each call to the subroutine.

Computation of the Trapezoid

Consider a missile at altitude DELZ above the terrain oriented as in Figure 2.4 from the side view with the axis of the sensor at angle SENANG from a normal to the missile flight path and the missile at the angle ANGMIS from the parallel to the terrain. The rearward field of view from the sensor axis is given by ALPHAR, and the forward field of view from the sensor axis is ALPHAF. The angle of the sensor to the missile SENANG, the forward angle of the field of view ALPHAF and the rearward angle ALPHAR are all input in common SCANNS. Subroutine BFLITE (as entry LNBFLT) computes the launch angle ANGLCH for each flight and the missile angle ANGMIS at each time T. Thus, if TYPMIS = 3 or 4, indicating that the sensor is fixed to the missile, the angle of the forward edge of the sensor field of view (FOV), ALPHFP is computed from

$$\text{ALPHFP} = \text{SENANG} + \text{ANGMIS} + \text{ALPHAF}$$

and the rearward edge ALPHRP from

$$\text{ALPHRP} = \text{SENANG} + \text{ANGMIS} - \text{ALPHAR}.$$

The angles ALPHFP and ALPHRP are determined in a vertical plane through the sensor. Alternately, if TYPMIS > 4 indicating a fixed sensor orientation depending only on launch angle these angles become

$$\text{ALPHFP} = \text{SENANG} + \text{ANGLCH} + \text{ALPHAF}$$

$$\text{ALPHRP} = \text{SENANG} + \text{ANGLCH} - \text{ALPHAR}.$$

When these values of ALPHFP and ALPHRP are determined the search variables can be computed.

The distance AL from the missile launch point to the rear edge of the field of view, using the new coordinate system with the flight path as the x-axis, can be found from the distance the missile has flown, RM1, plus the projection of the

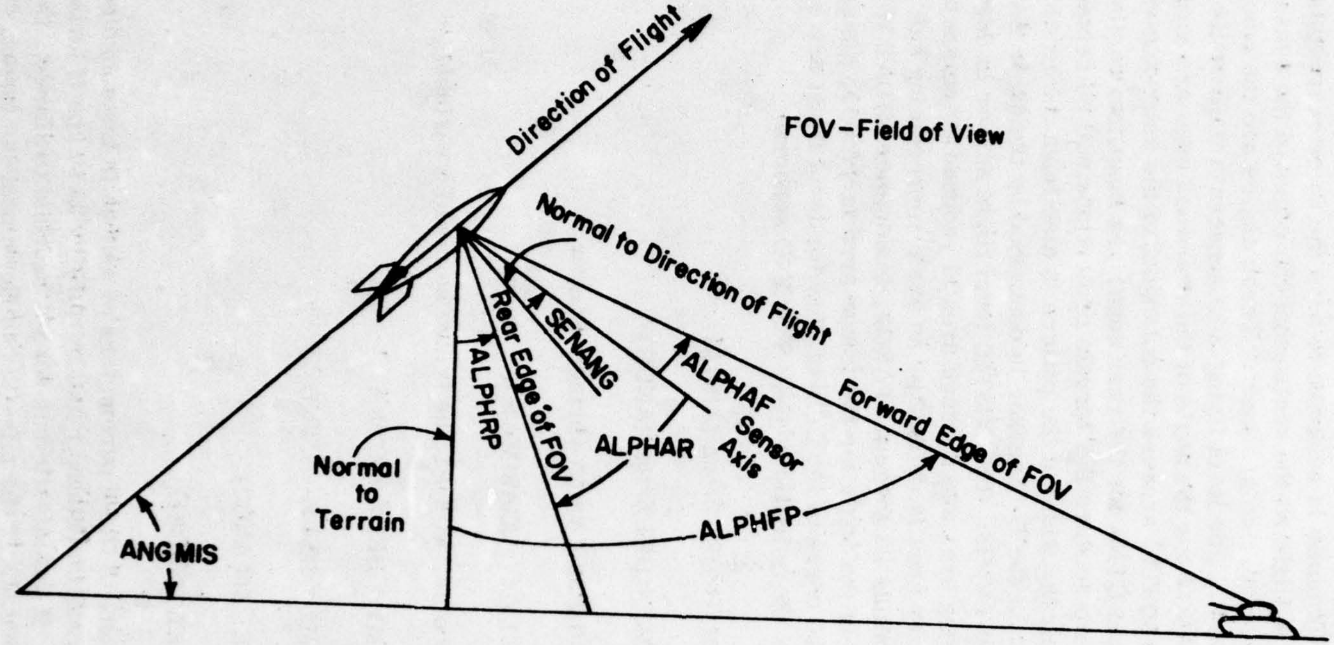


Figure 2.4.--Missile-Sensor Orientation

field of view computed from DELZ tan (ALPHRP). Similarly, the forward edge of the field of view FL is RM1 + DELZ tan (ALPHFP).

The level flight missile is assumed to have the same rectangular field of view during its entire flight so the angles that the edges of the sensor field of view make with a level terrain are input. These angles are the complements of ALPHFP and ALPHRP. For level flying E-O sensors ANGB for the angle of the forward edge and ANGA for the angle of the rearward edge are input into common TVMIS. For MISTIC sensors the cotangent of the same angles, CTNB for the forward edge and CTNA for the rear edge, are input into common MIDATA. For these two sensors the terrain projections will be rectangular and the angle describing the width of the pattern is also input. The width of half of the field of view of the E-O sensor is described by the angle GAMMA also input into common TVMIS. The MISTIC level flight sensor is described as having a primary sensing area and a fringe area of reduced detection possibility. The width of the primary area is defined by an angle representing half the angle of the field of view input as its tangent, TANG2, in common MIDATA. The corresponding tangent of the angle for the fringe area is TANG1, also input in common MIDATA. The edges of the field of view for level flight are computed in the same manner as the ballistic edges. For E-O sensors:

$$AL = RM1 + DELZ/\tan(ANGA)$$

$$FL = RM1 + DELZ/\tan(ANGB)$$

and the primary viewing half-width variable SL, as

$$SL = DELZ \tan(GAMMA).$$

For MISTIC these become, with PL the fringe half-width variable

$$AL = RM1 + DELZ \cdot CTNA$$

$$FL = RM1 + DELZ \cdot CTNB$$

$$SL = DELZ(TANG2)$$

$$PL = DELZ(TANG1).$$

The development of these parameters of search in the coordinate system of the missile flight path facilitates rapid initial checks to see if further computations are merited. If the missile has not yet reached the target, the procedures can be quickly terminated. If the missile has flown past the target, this too can be quickly found. Only when the target is within the forward and rear edges of the field of view is it necessary to check further by processing side angles.

Level Flight Search Procedure

The target is at distance XP from the launcher along the flight path. The forward edge of the field of view is at FL. If FL < XP, then the missile field of view has not yet reached the target and processing can be terminated until later. If FL ≥ XP then it will be necessary to determine if the target is within the rectangle or trapezoid of the sensor scan area on the terrain. This section describes the rectangular area search procedures.

Once FL ≥ XP, the target is in the fringe area of a rectangular sensor whenever SL < AYP < PL and when AYP < SL the target is within the primary sensor area. When either of these two conditions occur for level flight missiles the target is within the sensor's field of view and this condition will be identified for FLIGHT. Three unique cases require individualized processing when the target is in a rectangular scan area. These cases are:

1. First iteration when the missile begins to search (LD in common MIDATA equals 0).

2. Subsequent to the first iteration but the first time the target is in the field of view (LD > 0 and NOBRAK is .FALSE.)

3. After previously having been in the field of view (NOBRAK is .TRUE.).

Cases 1 and 3 will result in SEEKER setting IGUIDE to 2, so that FLIGHT can process the target as being in the field of view. In addition to setting IDUIDE, SEEKER records the following variables for subsequent processing by FLIGHT and FINAL or FINALE. These variables are:

L = flag to indicate if in primary or fringe area, set to 1 for primary area, zero for fringe area (in common OPEN)

DPL = width of fringe region, in common OPEN
= PL - SL

DYP = distance of the target from the fringe, in common OPEN
= PL - AYP.

The variables DPL and DYP are used in the Monte Carlo procedure by FINAL to determine whether the missile actually acquires the target.

For case 2 above, subroutine FLIGHT implements a recursive procedure in order to determine the time when the target first entered the missile field of view. This procedure is established in order to permit less frequent checks

prior to the target entering the field of view and to represent the entire time interval that the target is exposed to detection. A target can either enter the field of view by penetrating the front side of the rectangle that is perpendicular to the direction of flight or from one of the lateral sides parallel to the flight path. The time when the target enters the field of view is considered as being determined when the target is placed within a distance of EPS of one of the sides. That is, the target is on the front side when

$$FL - EPS \le XP \le FL \text{ and } AYP \le PL, \text{ and}$$

on a lateral side when

$$XP < FL \text{ and } PL - EPS \le AYP \le PL.$$

EPS is recorded in common MIDATA. The target penetrates the front side when

$$XP \le FL$$

$$FLP < XP$$

$$AYP < PL,$$

where FLP is the previous value of FL. FL is recorded in common OPEN. After penetration, the time of penetration is estimated by

$$B1 = B1 \cdot (1 - (FL - XP) / (FL - FLP)).$$

Recall that the missile flight time T is determined using B1 to interpolate within a time interval. Also, the target penetrates a lateral side when

$$XP < FLP$$

$$XP < FL$$

$$AYP < PL.$$

In this case, the time of penetration is estimated by

$$B1 = B1 + (PL - AYP) / (PLP - YPP) * (B1 - B1P)$$

where PLP, YPP, and B1P are the previous values of PL, AYP, and B1, respectively. PL, AYP, and B1 are recorded in commons MIFO, MIDATA, and LNSET, respectively. After identifying case 2 and computing a new value for B1, SEEKER sets ID to two to initiate the recursive procedure to determine the time of penetration by subroutine FLIGHT.

The field of view of a level flying missile may fly past the target or the target may move out of the possible field of view without the target being seen. When this occurs the condition observed will be:

$$XP < AL.$$

This condition indicates that the missile will not be able to detect the target at any later time and SEEKER returns this fact to FLIGHT by setting IGUIDE to one for termination of the missile as a simulation element.

Ballistic Search Procedures

The problem of evaluating whether a target for a ballistic missile is within the field of view is somewhat more complicated due to the trapezoid having two nonparallel sides. The trapezoid is assumed to have its parallel sides normal to the flight path. Thus, if the forward edge of the trapezoid is FL and the rear edge AL, the target is in the field of view only if $AL \leq XP \leq FL$. The complications arise when determining whether the target is between the two nonparallel sides. For rectangular search the lateral sides of the search area are parallel at a distance SL or PL to the flight path and the value of AYP provides a rapid check. For ballistic flight the sides may be at any angle to the flight path, complicating the search procedures. This complication is sufficient to merit checking the nonparallel sides only if the target is between the front and rear sides. The ballistic procedure then first compares XP with AL and FL. If $AL \leq XP \leq FL$, the lateral sides of the trapezoid will be checked. If $XP < AL$ or $XP > FL$, the process is terminated until the next call to subroutine SEEKER.

The trapezoidal search of a ballistic missile must first define the edges of the area to be searched. These edges are constructed by considering the trapezoid as the base of a pyramid with its apex at the sensor. A top view of this pyramid is shown in Figure 2.5. Consider the surface angles DELTAL and DELTAR shown in Figure 2.5 which define the orientation of the sensor's field of view. These angles are defined by the following three lines in the front surface of the pyramid:

1. the normal to the intersection between the terrain and the front pyramid surface,

2. the intersection between the front and left pyramid surfaces, and

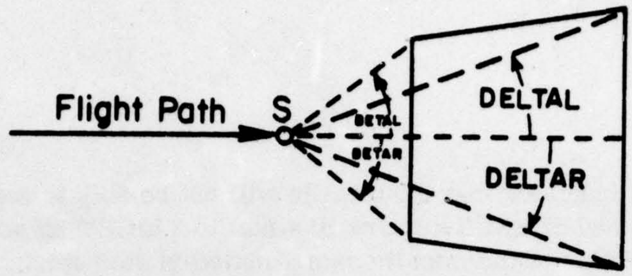


Figure 2.5.--Top View of the Sensor-Trapezoid Orientation

3. the intersection between the front and right pyramid surfaces.

DELTAL is the angle included between lines 1 and 2 above, and DELTAR is the corresponding angle between lines 1 and 3. The rearward angles are defined in the same manner in the rear pyramid surface to be BETAL to the left and BETAR to the right. These angles are input in common SCANNS to represent the sensor system. As the missile flies roll may develop in the system. The roll angle, ROLL, is assumed to be normally distributed around the center line, so the sensor angles will have small errors in their actual orientation. The standard deviation of the roll is input as ROLLRT in common MIDATA and the autocorrelation between the roll at successive points in time is input as RHO also in MIDATA. Each point in time is separated by a time interval of DELT time units in duration, and DELT is recorded in common MIDATA. The previous amount of roll added to the sensor is stored in RMEAN in common LNSET. The current roll for DELT time units later is computed as a zero mean Gaussian or normal autocorrelated process by a Monte Carlo operation as

$$\mathrm{CROLL} = \mathrm{RMEAN} * (\mathrm{RHO}) + \mathrm{ROLLRT} * (\mathrm{NORMAL}) * (1 - \mathrm{RHO})$$

where NORMAL is a random normal deviate, N(0,1), with a mean of zero and a variance of 1. For times within the time interval of length DELT, a linear interpolation between RMEAN and CROLL using B1 is employed. The interpolation for the roll, ROLL, at time T specified by B1 is performed by

$$\mathrm{ROLL} = (1 - \mathrm{B1}) \cdot \mathrm{RMEAN} + \mathrm{B1} \cdot \mathrm{CROLL}$$

The roll is added to the right side variables and subtracted from the left, or

$$\text{RETAR} = \text{BETAR} + \text{ROLL}$$

$$\text{RELTAR} = \text{DELTAR} + \text{ROLL}$$

$$\text{RETAL} = \text{BETAL} - \text{ROLL}$$

$$\text{RELTAL} = \text{DELTAL} - \text{ROLL}.$$

With roll considered, the procedure can now check the possibility of detection using the side angles.

The procedure to test for a target between the lateral sides of the field of view evaluates first the right side and then the left. If the target is further right than the right side of the trapezoid then it cannot be in the field of view. If it is left of the right side of the terrain trapezoid, then the left side of the trapezoid must be checked. The offset of the rear corner of the right side of the field of view, RFW, (see Figure 2.6) can be found from BETAR and the slant altitude corresponding to BETAR,

$$\text{DELR} = \sqrt{\text{DELZ}^2 + (\text{AL} - \text{RM1})^2}$$

$$\text{RFW} = \text{DELR} \cdot \tan \text{BETAR}.$$

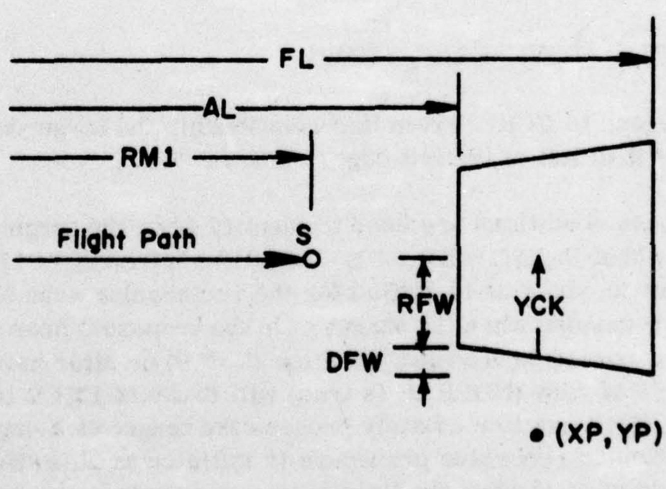


Figure 2.6.--Checking for a Target within the Trapezoid

The difference between this distance and the width of the front right side field of view DFW is then found for DELF the slant altitude correspondent to DELTAR

$$DELF = \sqrt{DELZ^2 + (FL - RM1)^2}$$

$$DFW = DELF \cdot \tan DELTAR - RFW.$$

The fraction of the distance from the rear edge to the target FRAC along the flight path is given by

$$FRAC = (XP - AL)/(FL - AL).$$

The distance from the flight path toward the target YCR on a line normal to the flight path intersecting a lateral side of the trapezoid is illustrated in Figure 2.6 and can be found from

$$YCR = -RFW \quad -FRAC \quad (DFW)$$

where YCR is negative to reflect that the point is to the right of the flight path. Thus, if the target is right of the right edge of the trapezoid then $YP < YCR$ and no detection can occur. If it is left then the left edge must be checked.

The left side is checked in the same manner using BETAL and DELTAL and computing

$$YCL = +RFW + FRAC \quad (DFW)$$

which is analogous to YCR. From this relationship the target is outside the field of view if it is left of the left edge, or if $YP > YCL$.

The above conditions are used to identify when the target is in the missile field of view. That is, $AL \le XP \le FL$ and $YCR \le YP \le YCL$. However, the same three unique cases as identified for the rectangular scan area require individualized processing when the target is in the trapezoid scan area. Thus, a target in the trapezoid on the first iteration (ID = 0) or after having previously been in the field of view (NOBRAK is true) will cause SEEKER to set IGUIDE to two so that FLIGHT can immediately process the target as being in the field of view. In addition, a recursive procedure is initiated to place the target on the edge of the field of view when the field of view moves to cover the target position (LD > 0 and NOBRAK is .FALSE.). ID is set to two by SEEKER to initiate this recursive procedure. When the target penetrates the front edge of the trapezoid, B1 is set as specified below.

$$B1 = (1 - (FL - XP) / (FL - FLP)) \cdot B1.$$

Also, the target penetrates one of the lateral trapezoid sides when $XP < FLP$. In this case,

$$B1 = B1 + (YER / YERP) \cdot DB1,$$

where

$$YER = \begin{cases} YP - YCR & \text{if the target penetrated the right lateral side} \\ YCL - YP & \text{if the target penetrated the left lateral side} \end{cases}$$

YERP = the value of YER on the previous iteration.

This recursive procedure continues until

$$FL - EPS \le XP \le FL$$

or

$$YCR \le YP \le YCR + EPS$$

or

$$YCL - EPS \le YP \le YCL.$$

Once the above conditions are identified, SEEKER sets IGUIDE to two so that target acquisition is possible.

The possibility of different sensor orientations on the missile requires continual checking by SEEKER during the flight. The fixing of the sensor to the missile permits the possibility of looking back toward previously searched ground. Thus, $XP < AL$ does not guarantee a fly-by, and SEEKER must be called until missile impact.

Summary of Computations in Subroutine SEEKER

Subroutine SEEKER performs the search operations for the missile system, and determines when the target enters the field of view or when the missile has flown past the target. The results of the sensor scan of the terrain relative to the target is returned to subroutine FLIGHT, from where FINAL (or FINALE) is called if the target is in the field of view. Subroutine FLIGHT also performs the bookkeeping of terminating the flight if the missile has overflown the target. When the missile is ballistic and no target appears in the field of view, or when any terminally guided missile loses lock-on, then FLIGHT will use the assessment portion of the artillery module to determine terminal effects upon missile impact.

Subroutine SEEKER determines the type of sensor on the missile, finds the orientation of the target to the field of view, and returns the appropriate value of IGUIDE to FLIGHT depending upon whether, 0) the missile field of view is still out of range, 1) the missile has flown past the target, or 2) the target is in the field of view. Computations for SEEKER are as follows:

1. The previous iteration variables, PL, DB1, B1P, and YPP are calculated by
   PLP = PL
   FLP = FL
   DB1 = B1 - B1P
   B1P = B1
   YPP = AYP

2. Target positions at time T are determined by calls to SPOT1 and SPOT2 and stored in (XT, YT).

3. Calculate the distance between launch point and target
   R = RGXY(XLP, YLP, XT, YT).

4. Calculate the angle between the missile direction of flight and a line from the launch point to the target
   THETA = ATAN2(YT-YLP, XT-XLP)-ATAN2(YM-YLP,XM-XLP).

5. Determine coordinates of the target relative to the flight path
   XP = R*cos(THETA)
   YP = R* sin(THETA).

6. Determine horizontal distance from the target to the missile flight path
   AYP = | YP |.

7. Determine missile height above target, DELZ = ZM1 - ZT.

8. If the missile has rectangular seeker, i.e., TYPMIS(ILNNUM) < 3, go to 17.

9. If the missile has E-O sensor, i.e., TYPMIS(ILNNUM) is even, go to 16.

10. The missile has a trapezoidal scan pattern. If the missile has a sensor with a constant orientation to the terrain, i.e., TYPMIS(ILNNUM) ≥ 5, go to 12.

11. The missile has a sensor attached to the missile, oriented with angle ANGMIS. Compute forward and rearward sensor angles,
ALPHRP = SENANG + ANGMIS - ALPHAR
ALPHFP = SENANG + ANGMIS + ALPHAF,
then go to 13.

12. The missile has a constant orientation to terrain
ALPHRP = SENANG + ANGLCH - ALPHAR
ALPHFP = SENANG + ANGLCH + ALPHAF.

13. Find the tangents of the sensor angles
TANARP = tan (ALPHRP)
TANAFP = tan (ALPHFP).

14. Set forward and rear search limits for the sensor
AL = RM1 + DELZ * TANARP
FL = RM1 + DELZ * TANAFP°

15. Set flag for trapezoidal search pattern, i.e., SL = 0, then go to 18.

16. The missile has an E-O sensor, set search limits
FL = RM1 + DELZ tan (ANGB)
AL = RM1 + DELZ tan (ANGA)
PL = DELZ * tan (GAMMA)
SL = PL,
go to 18.

17. The missile has a level flight, rectangular sensor, set search limits
AL = RM1 + DELZ · CTNA
FL = RM1 + DELZ · CTNB
PL = DELZ * TANG1
SL = DELZ * TANG2.

18. If target is not yet in field of view (XP > FL), then go to step 22.

19. If the missile has a trapezoidal search pattern, i.e., SL = 0, go to 31.

20. If the target is in the rectangular field of view, i.e., if AYP < PL, go to 23.

21. If the missile field of view passed beyond target, XP < AL, go to 30.

22. Return to FLIGHT for another search iteration and continue recursive procedure, set IGUIDE = 0.

23. If the target is in fringe region, i.e., AYP > SL, set L = 0, if not, L = 1.

24. The target is within the rectangle. If LD = 0 or XP ≥ FL - EPS or NOBRAK is .TRUE. or AYP ≥ PL-EPS, go to step 29.

25. If XP < FLP, go to step 27.

26. The target is penetrating the front edge. Set interpolation factor B1 = (1 - (FL-XP) / (FL-FLP))B1. Go to step 28.

27. The target entered field of view from the side. Set interpolation factor B1 = (1 - (PL-YP) / (PL-PLP))B1.

28. Set LD = 2 to initiate recursive procedure in FLIGHT for determining position where target penetrated field of view. Go to step 22.

29. Missile is in the field of view. Set IGUIDE = 2, LD = 1, DYP = PL-YP, DPL = PL-SL, and return to FLIGHT.

30. Note that the missile flew past the target. Set IGUIDE = 1 and return to close out flight.

31. Missile has a trapezoidal search pattern. If XP < AL, go to step 22.

32. Compute roll
ROLL = (1 - B1) · RMEAN + B1 · CROLL.

33. Determine right sensor limiting angles
TANBR = tan (BETAR + ROLL)
TANDR = tan (DELTAR + ROLL).

34. Compute distance to right rear sensor limit

$$DELR = SQRT(DELZ^2 + (AL - RM1)^2)$$
$$RFW = DELR * TANBR.$$

35. Compute difference between front and rear sensor limits

$$DELF = SQRT(DELZ^2 + (FL - RM1)^2)$$
$$DFW = DELF * TANDR - RFW.$$

36. Determine fraction of distance of target into field of view

$$FRAC = (XP - AL)/(FL-AL).$$

37. Determine edge of sensor field opposite target

$$YCR = -RFW - FRAC * DFW.$$

38. Determine left side sensor limiting angles

$$TANBL = \tan(BETAL - ROLL)$$
$$TANDL = \tan(DELTAL - ROLL).$$

39. Compute distance to left rear sensor limit

$$RFW = DELR * TANBL.$$

40. Compute difference between front and rear sensor limits

$$DFW = DELF * TANDL - RFW.$$

41. Determine edge of sensor field opposite target

$$YCL = RFW + FRAC * DFW.$$

42. Determine the current and previous values of target lateral error

$$YERP = YER$$
$$YER = YP - YCR$$
$$YEL = YCL - YP$$

If $|YER| > |YEL|$, then YER =YEL.

43. Check for target being within lateral trapezoid sides.

If YP < YCR, go to step 22.
If YP > YCL, go to step 22.

44. Target is within the trapezoid.

If LD = 0, or XP ≥ FL-EPS, or YP < YCR+EPS, or YP ≥ YCL-EPS or NOBRAK = .TRUE., go to step 29.

45. Recursive procedure is initiated to determine position where target penetrated field of view. Decrement missile flight time so that later increment by FLIGHT will give correct time.

T = T - DELT * B1
If XP < FLP, go to step 46, otherwise, go to step 26.

46. Target is penetrating from the side
B1 = B1 + (YER/YERP) · DB1, go to step 28.

Subroutine NUTARG

The preparation of this ballistic flight model occurred in conjunction with the effort to improve the supporting fire request procedure as reported in Chapter 3. The improved logic in fire support coordination and forward observer control of indirect flights was concurrent with developments of BFLITE and SEEKER to assure final compatibility. One interaction problem arose which was solved by a new subroutine, NUTARG. The original DYNCOM simulation used a routine NEWTRG in FLIGHT to pick a new target by an FO while the missile was in flight. This old routine was not compatible with the improved fire request procedure so a new routine, NUTARG, was designed to replace NEWTRG when used with the new firing logic.

Subroutine NUTARG is called during the first flight segment of an indirect fire mission. The launch of a missile occurs after time has been expended in communication, launcher loading, missile warmup, and other activities. During this time, targets available to the FO may change. The FO requesting an indirect fire missile gives the target coordinates, and the actual target element is not needed until flight. Thus, during the initial phase of indirect fire, subroutine NUTARG picks the highest priority target for the FO in the designated target area, if possible. When an indirect fire mission is initialized, the target number, ITARG, is set to zero. The first time the missile becomes current, NUTARG examines unassigned targets in the impact area detected by the FO and assigns the missile the highest priority target that is unassigned. In the event that all targets are assigned, then the missile is assigned the highest priority target available.

The procedure in NUTARG begins by finding the unit weapon code, KK from LWCOD in common LWCOD. The entire enemy element list is then cycled through, checking for two things. If the element is already dead, then it is ignored. The proximity criterion for the evaluation of the closeness of the enemy element location to the target coordinates (XAT, YAT in common OPEN) is contained in array CLOSE(KK) in common CLOSE. If the element is closer to (XAY, YAT) than CLOSE(KK) then it is a possible target. For all those that meet the closeness criterion the priority of the target for weapon code KK is determined using a call to routine APRIOR. The call to routine APRIOR checks intervisibility of the FO and the target, and returns a value of ITWT greater than zero if the target is detected and intervisible. If APRIOR returns a zero priority, the target is not processed further. When a nonzero value of ITWT is returned,

the target element number is put in LIST(IL) and the priority is placed in LWTL(IL). Each potential target is then checked in order of priority to find the entry with the highest priority that has no current missile assignment (no entry in MISAVE(1,K) where K is the missile number). When a review of all potential targets finds no possible element without an entry in MISAVE(1,K), then the element with the highest value in LWTL is assigned as the target for the missile.

The computations in NUTARG are summarized below.

1. Set unit weapon code, KK, to LWCOD(NUMELE + NUMART + NMISUN(NUM)), and variable IL to zero.

2. Set index I to first enemy element IFEELE.

3. If a dead element, LKILL(I) ≥ 3, go to step 8.

4. Find location of element I, CALL ELOC(I, XI, YI).

5. Find proximity of element to aimpoint. If XI - XAT > CLOSE(KK), or if YI - YAT > CLOSE(KK), go to step 8.

6. Find priority of element, ITWT. CALL APRIOR(I, NFO, ITWT, KK) and if ITWT = 0, go to step 8.

7. Enter element I on possible target list, set IL = IL + 1, LIST(IL) = I, and LWTL(IL) = ITWT.

8. Increment I. If I is not greater than last enemy element, ILEELE, go to step 3.

9. If no possible targets, i.e., IL = o, then return.

10. Initialize variables for search for highest priority target, I = 1.

11. J = I + 1
    MAXJ = I
    MAXPRI = LWTL(I)
    ISAVE = LIST(I)
    If there are no more targets on the list, i.e., if IL = 1, go to step 18.

12. Determine if target J is lower priority, i.e., if LWTL(J) < MAXPRI, go to step 14.

13. Record higher priority target
    MAXJ = J
    MAXPRI = LWTL(J)
    ISAVE = LIST(J)

14. J = J + 1
    If there are more targets to be checked, i.e., if J ≤ IL, go to step 12.

15. Search missiles to determine whether target ISAVE has already been assigned. K = 1.

16. Has ISAVE been assigned as a target for missile K? i.e., if ISAVE = MISAVE(1,K), go to step 19.

17. K = K + 1
    If there are more missile targets to check, i.e., if K ≤ NMSCLK, go to step 16.

18. Record ISAVE as current missile target, i.e., ITARG = ISAVE. Go to step 23.

19. If a higher priority unassigned target was not found, i.e., if MAXJ = I, go to step 21.

20. Put higher priority target in earlier position of list
    LWTL(MAXJ) = LWTL(I)
    LIST(MAXJ) = LIST(I)
    LWTL(I) = MAXPRI
    LIST(I) = ISAVE.

21. I = I + 1
    If there are more targets to be searched for highest priority unassigned target, i.e., if I < IL, go to step 11.

22. All elements in the list have been assigned as missile targets. Take top priority target and assign it as the missile target. ITARG = LIST(1).

23. Computations are complete.

VARIABLE DEFINITION INDEX

Variable	Definition	Variable	Definition
AL	p. 2-14	KK	p. 2-28
ALPHAF	2-14 (input)	L	2-17
ALPHAR	2-14 (input)	LD	2-17
ALPHFP	2-14	LIST(IL)	2-29
ALPHRP	2-14	LWCOD(NELE)	2-28
ANGA	2-16 (input)	LWTL(IL)	2-29
ANGB	2-16 (input)	MISAVE(1,K)	2-29
ANGLCH	2-5	NOBRAK	2-17
ANGM	2-9	NORMAL	2-20
ANGMIS	2-8	PL	2-3
ANG2	2-3	PLP	2-18
AYP	2-13	RELTAL	2-21
BETAL	2-20 (input)	RELTA	2-21
BETAR	2-20 (input)	RETAL	2-21
B1	2-7	RETAR	2-21
B1P	2-18	RFW	2-21
CLOSE(KK)	2-28 (input)	RHO	2-20
CROLL	2-20	RM(J)	2-7 (input)
CTNA	2-16 (input)	RMEAN	2-20
CTNB	2-16 (input)	RMO	2-5
DELF	2-22	RM1	2-6
DELR	2-21	ROLL	2-20
DELT	2-20 (input)	ROLLRT	2-20 (input)
DELTAL	2-19 (input)	SENANG	2-14 (input)
DELTAR	2-19 (input)	SL	2-3
DELZ	2-13	T	2-5
DFW	2-22	TANG1	2-16 (input)
DPL	2-17	TANG2	2-16 (input)
DRW	2-22	TD(J)	2-8 (input)
DYP	2-17	RF(J)	2-8 (input)
EPS	2-18 (input)	THETA	2-13
FL	2-16	TIMP	2-10
FLP	2-18	TM	2-13
FRAC	2-22	TT	2-13
g	2-6	TYPMIS(ILNNUM)	2-7 (input)
GAMMA	2-16 (input)	VM(ILNNUM)	2-8 (input)
IGUIDE	2-12	VM1	2-8
ILNNUM	2-6	XAT	2-10
ITARG	2-13	XM	2-9
ITWT	2-28	XP	2-13

Variable		Definition
XT	p.	2-13
YAT		2-10
YCL		2-22
YCR		2-22
YER		2-23
YERP		2-22
YM		2-9
YP		2-13
YPP		2-18
YT		2-13
ZM(J)		2-7
ZMD(J)		2-7
ZLP		2-8
ZM1		2-6
ZT		2-13
Z1		2-7
ZZ		2-7

REFERENCES

1. Hutcherson, D. C. and S. H. Parry, "Indirect-Fire Semi-Active Missile Module (U)," Chapter 1 (in) The Land Combat Model (DYNCOM), Missile and Electronic Countermeasures Models (U), Secret, edited by G. Clark, et al., RF 2376 FR 70-5 (S), Systems Research Group, The Ohio State University, January 1971, AD 517621.

2. Davis, Leverett, Jr., James W. Folling, Jr., and Leon Blitzer, The Exterior Ballistics of Rockets, D. Von Nostrand Company, Inc.: Princeton, New Jersey, 1958.

3. Merrill, Grayson, Principles of Guided Missile Design, D. Von Nostrand Company, Inc.: Princeton, New Jersey, 1956.

4. Dow, Richard B., Fundamentals of Advanced Missiles, John Wiley and Sons, Inc.: New York, 1958.

5. Demmy, S., "Indirect-Fire Ballistic Weapons," Chapter 10 (in) The Land Combat Model (DYNCOM), edited by A. B. Bishop and G. M. Clark, RF 2376 FR-1 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

CHAPTER 3

SEMIACTIVE MISSILE LAUNCH OPERATIONS

by

G. M. Clark
R. J. Wilhelm .

Introduction

As described in Chapter 1, two methods of delivering semi-active point-target or MISTIC missiles are simulated in DYNCOM, i. e., direct and indirect. This chapter describes the procedures for representing the missile launcher fire-support element in the indirect-fire mode for both helicopter and ground weapons. Moreover, the entire launch procedure for ground weapons in both direct and indirect-fire modes is presented in this chapter. Reference 1 describes the techniques simulated for the forward observer, FO, to select MISTIC missiles as the weapon of choice for an indirect-fire assignment, and the procedures necessary for the FO to locate the target or target complex for the launcher. Reference 2 describes the communication processing for these indirect-fire requests through the MISTIC Unit Fire Controller which is similar to the Artillery Fire Direction Center. Chapter 9 describes the selection of MISTIC as a direct-fire weapon of choice and the establishment of the initial launch request for direct-fire missions. This chapter begins with the receipt of the fire request at the launcher, and the accomplishment of the procedures requisite to simulate a launch. Flight of the missile is accomplished using the existing FLIGHT routines of DYNCOM. (See reference 1 and Chapter 2 for a description of the flight phase.) The procedures required to fire MISTIC and other ordnance from helicopters will be reported in Chapters 6 and 8 as these procedures have been specifically developed to simulate helicopter activities.

The launch of MISTIC missiles from any source in either fire mode was initially designed and reported (reference 3 to be accomplished in a single subroutine, i. e., LAUNCH. When fire-support coordination procedures were developed for aerial, artillery, and MISTIC fire support; this single routine was found to be inadequate and an effort to develop launching procedures consistent with these improvements was undertaken. This chapter reports the new launch procedure. It will be assumed that fire requests initiate as reported in Chapter 9 and reference 1, and communication of indirect-fire requests to the MISTIC Unit Fire Controller as reported in reference 2 are familiar to the reader. The procedures herein will launch a missile toward a target. The current procedures to simulate the flight of the missile to the target, target acquisition, and impact or flyby are reported in Chapter 2.

As indicated in Chapter 1, there are two modes of fire represented in DYNCOM for supporting units: direct and indirect. Direct fire is the launching of a missile toward a target that has been detected by the launcher prior to launch and which is in the launcher's field of view. The process simulated for direct fire uses direct contact between firer and requester, and direct sighting of the target. In this case, necessary communications are over tactical radio nets used to control units in contact with the enemy. Conversely, indirect fire is made in response to requests by FO's through fire-request communication channels. These channels must be used to request fire, locate the target, and verify that a target still exists when the launcher is ready to fire. Since the FO is providing the eyes for the launch crew, the communication of information prior to launch is essential, and busy communications nets influence launcher reaction in a manner differently than in the direct-fire situation where the target can be seen by the crew. As described in Chapter 1 and explained in reference 1 and Chapter 6, the process of selecting weapons for targets and communicating the request to launcher has been separated into four different sets of procedures, i. e., direct fire by ground elements, direct fire by helicopters, indirect fire by ground launchers, and indirect fire by helicopter launchers. Each of these sets of procedures uses different routines to represent preparation for luanching a MISTIC weapon. In the direct-fire mode, launch preparation is represented by LAUNCH for ground vehicles and HLNCH for helicopters. For indirect fire, MFB represents the activities of the missile launcher fire-support element for both helicopter and ground elements. That is, MFB represents communication between the launching vehicle and the requesting FO. Indirect-fire missile preparation for helicopter launches is represented by HLNCH. For ground launches, missile loading, set up, and warmup in the indirect-fire mode is represented by LAUNCH. The launching by ground elements and the indirect-fire missile fire support element for both ground and aerial vehicles are described below. A description of MISTIC missiles fires by helicopters is incorporated with other helicopter models and reported in Chapters 6 and 8. In all cases, however, the final result of a launch event is a missile launched toward a target to be assessed by the flight routines of Chapter 2.

Design Criteria

Before beginning a detailed discussion of the launch routine, the design criteria for simulating MISTIC flight will be reviewed briefly. The original semiactive missile development was undertaken for the U.S. Army Missile Command to assist in the evaluation of MISTIC missiles in land combat.

Design criteria were established for the original indirect-fire missile module to provide guidance to the missile representation. The sequence of events explicitly represented in the original MICOM module are repeated here:

1. target detection by the forward observer (FO);

2. target selection and fire request preparation by the FO;

3. communication of the fire request by the FO;

4. selection of a launcher which is available for mission;

5. launch preparations at the selected launcher;

6. transmission of the verification message of the launcher to the FO, if required;

7. missile launch;

8. missile flight--FO initiates illumination on the target;

9. target enters missile field of view;

10. acquisition of reflected illumination by the missile;

11. missile guided to target by reflected illumination; and

12. termination of missile flight (impact or flyby).

The simulation procedures, routines, and model interactions which were developed for the U.S. Army MICOM have been reported in reference 3. These basic routines were modified to include ballistic flight, and these simulation procedures are reported in Chapter 2; the following performance characteristics are simulated:

1. the accuracy with which the target, FO, and launcher can be located, possibly utilizing such equipment as a laser range finder;

2. distribution of the fire-request preparation times;

3. distribution of the fire-request transmission times;

4. verification message requirement;

5. launch time distribution;

6. ammunition supply and associated reload time distribution;

7. missile reliability;

8. nominal flight path, flight speed, and altitude;

9. accuracy of missile in traversing nominal flight path;

10. missile field of view (seeker geometry);

11. illuminator accuracy;

12. number of FO's illuminating a target;

13. length of time that FO illuminates a target;

14. intervisibility requirements;

15. ability of an FO to switch targets;

16. ability of the missile to acquire a signal;

17. susceptibility of the missile seeker to acquire the wrong signal;

18. control logic of the missile in tracking on a signal; and

19. missile accuracy and lethality.

The original assumptions and design criteria as presented in reference 3 have been retained in the present version. In addition, some new assumptions have been made to facilitate the current development effort and structure some interactions between direct fire, indirect fire, and movement. Listed below are the assumptions for ground weapon launches incorporated into the new module.

1. Launcher indirect-fire activities are suspended while engaging a direct-fire target and vice versa,

2. Launcher cannot fire a missle

   a. indirect while moving
   b. direct while moving unless user so specifies

c. while loading
d. while direct-fire missile is in the air
e. while indirect-fire EO missile is in the air,

3. Launcher cannot begin to load missiles

a. while moving
b. during a firing event
c. while a direct-fire missile is in the air
d. while an indirect-fire EO missile is in the air, and

4. A launcher cannot begin to move

a. during an indirect-fire event (for duration of salvo)
b. during a direct-fire event unless the user so specifies
c. while loading missiles
d. while a direct-fire missile is in the air unless the user so specifies
e. while an EO indirect-fire missile is in the air.

These design criteria and assumptions have been applied to the direct and indirect-fire launching procedures for ground weapons described below and the launching operations simulated. Since loading of missiles for ground launchers is common to both direct and indirect-fire, loading procedures are described first.

Missile Loading for Ground Elements

Ground missile launchers may carry a reserve supply of missiles that must be loaded prior to warming up a missile and launching. Multiple rounds can be loaded in each load cycle, and the standard number of rounds that can be loaded during any cycle is input in NRL(LCOD),* where LCOD is the MISTIC weapon code (see Chapter 1). The number of rounds in missile reserves available for loading is recorded by NM(LCHN), where LCHN is the element's MISTIC launcher number. The number of rounds loaded for each weapon and each projectile type is recorded in array LAMMO. Thus, an entry in LAMMO is incremented by a value up to NRL(LCOD) when MISTIC missiles are loaded, and the entry in the array LAMMO incremented is for the launcher conducting

---

*Most common areas in the routines described in this chapter have the same names as the variable names. When this is true, no common name will be mentioned.

loading and for the MISTIC projectiles.

During the firing procedures, loading of missiles occurs under two different conditions, i. e., routine loading when the launcher is not otherwise engaged, and loading up to a specified level to fire a mission. The first situation occurs between mission assignments when the launch crew is not otherwise engaged and other conditions for loading are satisfied. The second situation occurs when an indirect-fire mission request specifies more missiles than are currently loaded. Also, as noted in the above discussion, there are different places in the luanch procedures where loading of one type or the other occurs. To provide either type of loading whenever needed, subroutine LOADM was developed.

Subroutine LOADM is called with the launcher number, LCHN, the time available to load, DELT, and the number of rounds to be loaded, NRD. If the routine is to load as many as possible in time DELT, then NRD is set to zero. If the routine is to load up to NRD rounds without any constraint on time, then DELT is set to zero. During initialization in LOADM, the loading time spent, TIM, is set to zero and the number of rounds in reserve, N, is set to the storage location NM(LCHN). The number of rounds already loaded, LAM, is found from the array LAMMO. Other variables initialized are:

NR = number of rounds loaded at one time, input in NRL(LCOD),

NML = maximum number of rounds that can be loaded on launcher, input in NMLIM(LCOD),

TB = mean time to load one group of NR, input in TBL(LCOD),

SD = standard deviation of time to load, input in SDL(LCOD), and

TL = last computed load time from TB and SD in TLOAD(LCHN).

The time to perform one loading cycle and load NR rounds (or less), TL, is described as a normally distributed random variable with mean TB and standard deviation SD. Sometimes during the computational procedure a value for the load time TL is computed and not used. When this occurs, TL is recorded in TLOAD(LCHN) for later use. If TLOAD(LCHN) < 0. 0, then a new value for TL must be computed.

Two internal counters, NDEL and DELTT, are used to control the loading process as indicated by the calling parameters. NDEL counts the remaining

rounds to be loaded, and DELTT the remaining time to load. Each counter is decremented (by NR and TL, respectively) when a loading operation is completed, and the routine terminates when either counter expires (as shown below).

Thus, the first task in LOADM is to determine the loading constraints. If NRD = 0, then as many missiles as possible are to be loaded in time DELT. For this condition, NDEL is set to FINITY, a very large number, and DELTT to DELT, so the loading task will terminate when DELTT expires. If NRD $\neq$ 0, then NRD rounds are to be loaded, so NDEL is set to NRD. If DELT specifies a positive time limit for loading the NRD rounds, DELTT is initialized to DELT, otherwise it is set to FINITY and the loading task will terminate when NDEL reaches zero.

When these counters have been initialized, a computational cycle is performed where each cycle results in NR additional rounds being loaded; however, if the reserve is inadequate to load a full complement of NR missiles, i.e., N - NR < 0, then N missiles are loaded. Regardless of the number loaded during each cycle, a value of TL, the cycle load time, is generated using a normal distribution with mean TB and standard deviation SD.

The routine terminates when one of four conditions are satisfied.

1. Insufficient time remains to load NR rounds (DELTT $\leq$ TL);

2. NRD rounds have been loaded (NDEL $\leq$ 0);

3. The Launcher is full (LAM + NR > NML); and

4. No more reserves exist (N = 0).

When one of these conditions has been met the supply of loaded and reserve missiles is adjusted, and the actual number of rounds loaded and actual loading time consumed are returned in parameters NRD and DELT, respectively.

In order to maintain maximum readiness, LOADM is called by LAUNCH for each event that loading can occur. That is, a stationary nonfiring ground launcher is assumed to be loading until its maximum load NMLIM(LCOD) is reached. This process is simulated by LAUNCH calling LOADM to load missiles for the launcher's previous event if the launcher was stationary (EVBAR(ICE) = 0 where ICE is LCHN's element number), not firing (LFIRE(ICE) = 0), and not controlling a missile in flight (LFLAG(LCHN) $\neq$ 2). For this call to LOADM, the time interval during which loading could occur is required as the DELT calling parameter for LOADM. The duration of the launcher's previous event is noted by PETIM; however, the available loading time may also include time in the event just before the previous event during

which loading could not occur because a loading cycle not be completed. This unused loading time in an event because the event time was insufficient to complete a loading cycle is recorded in TUSLD(NUM). Thus,

$$\text{DELT} = \text{PETIM} + \text{TUSLD(NUM)}.$$

After LOADM is called, a new value of TUSLD(NUM) is determined by LAUNCH. Recall that the loading time used by LOADM is returned to the calling program in DELT. Thus, the new value of TUSLD(NUM) is determined by

$$\text{TUSLD(NUM)} = \text{PETIM} + \text{TUSLD(NUM)} - \text{DELT}.$$

If loading did not occur during the previous event because the launcher was performing activities such as firing or moving, then TUSLD(NUM) is set to zero regardless of its previous value.

Direct Fire from Ground Elements

Direct-fire activities by ground elements have been simulated extensively by DYNCOM. The last major modification (reference 3) added MISTIC missiles as another type of weapon to be simulated. This section reports the launch procedures for direct fire from ground vehicles developed to complement the other efforts described in this report.

Chapter 1 describes the basic procedures to be simulated for each ground vehicle to represent target acquisition and fire against those enemy units which are intervisible with the vehicle. These basic procedures are periodically performed for each active element in the simulation whenever the element becomes a current element.

As described in Chapter 1, each element has an entry in the basic clock array, ECLOCK. Whenever a launcher vehicle becomes a current element (has the lowest time in ECLOCK), the element represented by this vehicle is processed as would be any other element. During this processing, the vehicle will receive communications, detect targets, initiate communications, move if permitted, and if subroutine FIRCON selects a missile as the weapon of choice for a target and the launcher is able to fire during the current event a launch event is generated. The processes of target acquisition and communication of intelligence has been described in reference 4 under INTELL and COM, respectively. Movement is possible under some firing conditions and thus the interaction of movement with firing must be discussed below. Other movement procedures have been reported elsewhere (reference 5).

Figure 3.1 presents the overall flow of events involving direct fire by MISTIC missiles. The process starts with the launcher in an inactive state with respect to MISTIC direct-fire action where the launcher is searching for targets, reevaluating targets, loading missiles from the reserve supply, and other activities. Once a target is selected, the launcher may have to move to a fire position. Once he is in a fire position, he will have to load missiles from available reserves if there are none loaded. Once loaded, a missile will have to be set up and warmed up. A misfire may occur, and, if so, the misfire is removed. After each event, the fire controller (subroutine FIRCON) may change the target or ammunition selected and interrupt the process. Also, another MISTIC direct-fire event may be started after a successful launch.

The discussion below will assume that the procedures in subroutine FIRCON have initiated a request for a missile and that subroutine LAUNCH has been called to simulate the launch activities. The discussion below will initiate from the point where FIRCON (see flow chart in Volume 2) has set the flags KFIREV = 1 to indicate that firing is to occur and LNCH = 1 to indicate that a missile is to be launched. These two conditions indicate a missile launching event for a ground vehicle when the DYNCOM main program calls subroutine LAUNCH.

Initialization

The call to subroutine LAUNCH includes a calling parameter, TIME, to return the time spent in the launch activities. TIME is set by LAUNCH to represent the duration of the simulated procedures of loading, missile set up and launching. Added time delays for loading are required only when a missile is to be fired and insufficient missiles were loaded during previous nonfiring stationary periods. Two separate identification numbers are available to control which launcher is involved; ICE identifies the current element by its vehicle number in the list of all elements, and LCHN identifies the number of the launcher relative to other launchers. ICE and LCHN are both initialized during the identification of a new current element by subroutine GETICE and stored in COMMON/ICECOM/ for use by LAUNCH.

The identification of the present element in relation to its fire-support unit and its appropriate weapon code occur next. The fire-support unit that launcher LCHN belongs to is NT, found by:

NT = NUMART + NMISUN(LCHN)

where

NUMART	=	number of artillery units (in COMMON/NUMBER/)
NMISUN(LCHN)	=	MISTIC unit to which LCHN belongs (in COMMON/NMISUN/).

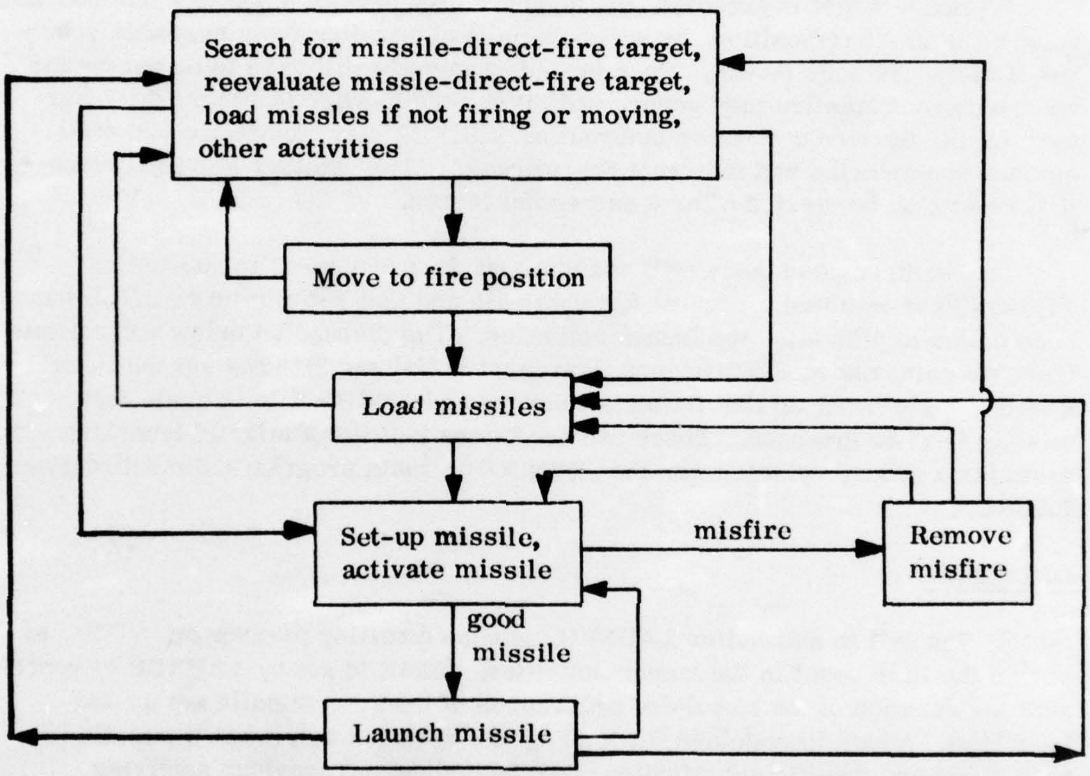
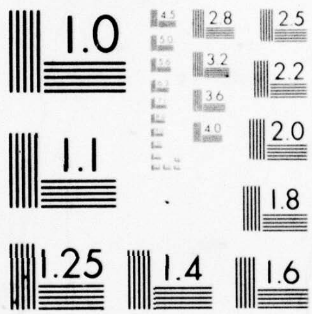


Figure 3. 1. --Launcher Direct-Fire Events

AD-A040 052 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL, (DYNCOM). VOLUME 1. HELICO--ETC(U)
DEC 71 G M CLARK DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-1(U) NL

2 OF 4
AD
A040052



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

NUMART is initialized during input to the basic data set. NMISUN is initialized as an array containing the missile unit identifiers. Weapon codes for weapon units were described in Chapter 1. For missiles, the weapon codes are obtained from:

LWC = fire support unit weapon code (see Chapter 1)
= LWCOD(NUMELE + NT), where NUMELE is the total number of elements (COMMON/NUMBER/).

LCOD = MISTIC weapon code (see Chapter 1)
= LWC - MWART, where MWART is the number of artillery units (COMMON/NUMBER/).

To calculate these variables, LWCOD, NUMELE, and MWART must be part of the basic input data set.

Direct-fire loading and launching activities can be suspended when the launcher already is controlling a missile in flight. The basic controls for launch, load, or suspend are the flag, LFLAG(LCHN) and the direct-fire launcher commitment clock TDFRDY(LCHN).

$$\text{LFLAG(LCHN)} = \begin{cases} 2 & \text{the missile launched by launcher LCHN is still flying and being controlled by LCHN,} \\ 0 & \text{otherwise.} \end{cases}$$

The basic purpose of LFLAG(LCHN) is to suspend additional direct-fire launchings while a previous direct-fire missile is still flying toward the target. When the missile is launched, LFLAG(LCHN) is set to two. When the missile impacts or flies by the target, LFLAG(LCHN) is set to zero by FLIGHT. The final initialization step is to check LFLAG(LCHN). If LFLAG(LCHN) is not two, the procedure checks to see if missiles are loaded. If LFLAG(LCHN) is two, then the direct-fire launch is delayed. Once LFLAG(LCHN) is set to zero by FLIGHT, the time of the fly by or missile impact is recorded by FLIGHT in the direct-fire launcher commitment clock, TDFRDY(LCHN). The significance of TDFRDY(LCHN) is that the launcher is committed to a direct-fire mission until the clock expires and this commitment prevents new direct-fire or indirect-fire actions.

Missile Loading

Following initialization, and determination of missiles loaded in LCHN's previous event, the missile loading status of LCHN is investigated. The current number of missiles loaded, ICNT, is determined by a call to AMMO using the parameters ICE and ITYP. If at least one round is loaded, then the launch can proceed. If no rounds are loaded, then loading will be needed before launch. The procedure first checks to see if the launcher is stationary; if not, launching

and loading are terminated for this current event. When the launcher is stationary, then the reserves are checked, and if not zero, then loading will occur. If no rounds are loaded, and none are in reserve, then the launch operation is terminated.

The loading procedure commences by first determining the time for the loading. The time required to load a standard cycle of NRL(LCOD) missiles is determined using the same procedure as employed by subroutine LOADM. The loading procedures next update the ammunition supply data. The number of rounds of missile reserves available for direct fire to the launcher is recorded in NM(LCHN). If NM(LCHN) $\geq$ NRL(LCOD), then NRL(LCOD) missiles are loaded; otherwise NM(LCHN) missiles are loaded. The number of rounds in reserve is decremented by the number of missiles loaded. The number of rounds loaded for each weapon, stored in the array LAMMO, is incremented by the number of missiles loaded.

Direct-Fire Launching

The launch procedure sets up the launcher, warms up the components before the actual launch, and tests the missile for failure. There is assumed to be a randomly varying setup time for slewing the weapon and laying-on the target which occurs between loading and missile warm up. This time will differ between the first and subsequent rounds, requiring longer on the average for the first laying operation at a given target. This time is expressed as a normally distributed delay with mean TBFR(LCOD) and standard deviation SFR(LCOD) for the first round to be launched, and TBSR(LCOD) and SSR(LCOD) for subsequent rounds. To control which of these two values are used, a flag LFRND(ICE) initialized by FIRCON indicates whether this is a first round. The launch procedures check LFRND(ICE) to see if this is the first missile set up. If LFRND(ICE) is equal to zero, then the missile has not been set up previously, this is the first round, and LAUNCH will set LFRND(ICE) to one. If it is equal to one, the missile has been previously set up and this is a subsequent round. After selecting the appropriate time parameters, TBFR and SFR or TBSR and SSR, LAUNCH Monte Carlo's for a set up time and this time is stored in the variable TIME.

LAUNCH next selects the actual missile to be fired and decrements the ammunition supply by decrementing the appropriate entry in the array LAMMO. The warm up time is input in WT(LCOD), and this time is added to TIME. The missile is then checked for failure. PNG(LCOD) is the input probability of failure of the missile. LAUNCH does a Monte Carlo comparison to determine if failure occurs. If the missile fails, the launch time accumulated (TIME) to this this point is incremented by the time required to process the failed missile, TDUD(LCOD), and the launch procedure is terminated until the next time the launcher becomes a current element. If the missile does not fail, then processing continues.

The final preparation of the missile for launch occurs only if the missile will be launched and is not a dud. This preparation begins by setting a dummy variable TM equal to the actual launch time, CLOCK + TIME. The element round count, LRNDC(ICE), used by FIRCON to control the length of the firing assignment, is decremented by one. The missile is created by assigning it a number, MISNUM, at time TM in subroutine CREATM. The missile will fly toward the target in the direct-fire mode, under the control of the launcher as the FO so all other activities are suspended by setting LFLAG(NUM) to two and setting the direct-fire launcher commitment clock, TDFRDY(LCHN), to TM. The missile will now be launched at time TM and flown by subroutine FLIGHT, but data for the flight must be transferred to subroutine FLIGHT.

Transfer of Data to FLIGHT

Subroutine MICONP takes the data for a current missile event and stores it in COMMON/MISAVE/. Later when a missile event occurs, subroutine MICONG retrieves the data from MISAVE. The procedure to use MICONP begins by placing missile flight parameters in commons /LNSET/ and /OPEN/. COMMON/LNSET/ is loaded as follows:

TLN = time of the launch
= TM,

XLN, YLN, ZLN = coordinates of the launch
= XE, YE, ELVATE(XLN, YLN, ICE),

ITARG = element number of the target
= LSTGT

NCE = element number of the launcher
= ICE,

LFO = element number of the forward observer directing the fire (the launcher for direct fire); i.e., LFO = ICE,

NUMM = the launcher number, LCHN,

IFO = the number of the FO relative to other FO's (the launcher number LCHN for direct fire),

T = time into the flight = zero, and

FINFLG = 1 to indicate a direct-fire event.

COMMON/OPEN/ is loaded as follows:

(XAT, YAT) last known coordinates of the target at launch time.

When these values are entered into the appropriate commons, subroutine MICONP is called and MISAVE is loaded for future flight events. At the same time, data must be loaded into COMMON/ICECOM/ to indicate a completion of the fire event. The target element number is set into variable LFELTK, and the time of launch put into EFELCK. The final step in the launch procedures is to increment TDFRDY(NUM) to the launch time. When this is accomplished, the direct-fire launch procedures are completed.

Direct Fire Computational Procedures in Subroutine LAUNCH

Subroutine LAUNCH contains the following computational steps for direct fire by ground launchers. Notice that after initialization and optional loading activities, a check is made in step 16 to determine if the launcher has a direct-fire target assigned. If so, direct fire processing continues, otherwise processing for possible indirect-fire activities is initiated at step 56. Step 56 and subsequent indirect-fire steps appear after a discussion of ground launcher vehicle indirect-fire activities later in this chapter beginning on page 3-35.

1. Define ICE as current element number, NUM and LCHN as the launcher number of the current element.

2. Determine NT, the fire support unit containing NUM; i. e., NT = NUMART + NMISUN(NUM).

3. Determine LCOD, the MISTIC weapon code for NT; i. e., LCOD = LWCOD(NT + NUMELE) - MWART.

4. Determine LWC, the weapon code for unit NT; i. e., LWC = LCOD + MWART.

5. Determine NF, the fire support firer number; i. e., NF = NUMART + NUM.

6. Determine NFBCLK, the clock number of the firer; i. e., NFBCLK = NF + NTFB.

7. Compute the launcher subscript KSUB; i. e., KSUB = ITOTFO + NUM.

8. Determine ITYP, the ammo code of the missile; i. e., ITYP = IFMC(LCOD).

9. Initialize the event duration time, TIME, to 0.

10. If NUM was moving (EVBAR(ICE) > 0), fired a missile or

conventional round (LFIRE(ICE) > 0), or was controlling an in-flight missile (LFLAG(NUM) = 2) in his previous event, then go to step 15.

11. Set DEL to the time available for loading in ICE's previous event; i.e., DEL = MIN(ETIM(ICE) + TUSLD(NUM), CLOCK - TDFRDY(NUM)).

12. If DEL is less than or equal to zero, then go to step 15.

13. Load as many missiles as possible in time DEL by calling subroutine LOADM with parameters LCHN = NUM, DELT = DEL, and NRD = 0.

14. Save any unused load time from the previous event by setting TUSLD(NUM) = DEL - DELT, then go to step 16.

15. Record any unused load time during the previous event; i.e., TUSLD(NUM) = 0.

16. If this launcher does not have a direct fire target assigned (LSTGT = 0), then go to step 56 on page 3-35 R.

17. If IFBMIS(NF) > 0 and TC(NFBCLK) < FINITY, then begin to suspend indirect-fire activity by resetting the firer's clock NFBCLK to CLOCK + EPSILN.

18. If the direct-fire activities of NUM have been suspended, set TIME to the standard event time and return.

19. If ICE is to fire this event, but not launch a missile, return.

20. If ICE is to launch a missile this event, go to step 22 and determine missile availability.

21. If ICE is moving to a fire position, return with TIME still set to zero; otherwise, NUM is controlling a missile, so set TIME = TDFRDY(NUM) - CLOCK + EPSILN and return.

22. Determine ICNT, the number of rounds of ammo ITYP that are loaded, by a call to AMMO(ICE, ITYP, ICNT).

23. If at least one round of this type is loaded, go to step 34.

24. If the launcher is moving (SPD > 0), then set IFIR and LNCH to 0, TIME = EVTIM(LWC), and return.

25. If the time to load has been determined, (TLOAD(NUM) $\geq$ 0), go to step 28.

26. Monte Carlo for the time to load, TL, from TL = TBL(LCOD + N(0, 1) * SDL(LCOD) where N(0, 1) is a normally distributed number with mean 0 and variance 1.

27. Go to step 29.

28. Set TL equal to time to load, TLOAD(NUM), reset TLOAD(NUM) to -1.

29. If a standard load of NRL(LCOD) rounds is available, load it; i. e., set NRLOAD = NRL(LCOD) and go to step 32.

30. If reserves are completely exhausted, delete the target by setting IFIR = 0, LNCH = 0, and TIME = EVTIM(LWC); then return.

31. Load the remaining reserve rounds; i. e., set NRLOAD = NM(NUM) and go to step 32.

32. Update the loaded and reserve ammunition supplies; i. e., decrement NM(NUM) by NRLOAD, increment LAMMO(ITYP, ICE) by NRLOAD.

33. Set TIME = TL, then return.

34. If the launcher was set up for firing before; i. e., this is not the first round, LFRND(ICE) > 0; then go to step 38.

35. Monte Carlo for time required to set up the first round, TSETUP, from TBFR(LCOD) + N(0, 1) * SFR(LCOD).

36. Set LFRND(ICE) to one to show the launcher is set up.

37. Go to step 39.

38. Monte Carlo for time required to set up a subsequent round, TSETUP, from TBSR(LCOD) + N(0, 1) * SSR(LCOD).

39. Initialize the event time, TIME, to TSETUP.

40. Select a missile and decrement the missile supply.

41. Increment the event time, TIME, by the missile selection and warm up time WT(LCOD).

42. Monte Carlo to see if missile is a failure by comparing random number to PNG(LCOD).

43. If not a failure, go to step 46.

44. Increment the event time, TIME, by reaction time for a misfire, TDUD(LCOD).

45. Set IFIR and LNCH = 0, then return.

46. Determine the missile launch time, TM, from CLOCK + TIME.

47. Decrement the round count, LRNDC(ICE), by 1.

48. Create a missile element, MISNUM, by call to CREATM(TM, MISNUM).

49. Set up missile data in COMMON/LNSET/ and COMMON/OPEN for transfer to subroutine FLIGHT:
   a. Set launch time TLN to TM;
   b. Set XLN, YLN to launcher location;
   c. Find launcher elevation, XLN, from ELVATE(XLN, YLN, ICE);
   d. Set target number ITARG to LSTGT;
   e. Find location of target; CALL ELOC(ITARG, XAT, YAT);
   f. Set launcher and FO numbers, NCE and LFO, to ICE;
   g. Set IFO to launcher number (NUM);
   h. Set flags: T to 0 and FINFLG to 1; and
   i. Set launcher number NUMM to LCHN.

50. Determine projected impact time TIMP, horizontal flight range RM1, and angle of launch ANGLCH (if NUM fires ballistic missiles); i. e., CALL LNBFLT.

51. Store the data now in COMMON/LNSET/ and COMMON/OPEN/ in COMMON/MISAVE/; i. e., CALL MICONP.

52. Record data for this event in COMMON/ICECOM/; i. e., LFELTK = LSTGT, EFELCK = TM.

53. Set the launcher availability time, TDFRDY(NUM) to CLOCK + TIME.

54. Suspend further direct-fire activities until impact by setting LFLAG(NUM) = 2.

55. Return.

Indirect Missile Fire

Indirect fire of missiles by either ground or aerial vehicles is represented as a different process than direct fire. The primary difference is that the launcher does not see the target and must depend on communications with the FO for location and verification of the target before the launching can occur. Following launching, the launcher is free to perform other operations including launching while the missile is in flight, unless the missile is of the special electro-optical, EO, type. A launcher can perform other combat activities in addition to indirect fire. This can be viewed as a dual role for the launcher, one as a combat vehicle in the direct-fire role, where it faces targets directly and fires, and one in the indirect-fire role, where it is remote from combat and fires against unseen targets upon request of the FO.

The time delays required to communicate between the forward observer and the missile launcher are an important characteristic of the indirect-fire process. Also, communication times can occur simultaneously with other combat activities; consequently, a separate element, the missile launcher fire-support element, has been created explicitly to represent these time delays. The activities of this element are represented by subroutine MFB for both helicopter and ground launchers. The missile launcher fire-support element has its own clock to sequence its events by the main program along with the combat elements.

Launch preparation activities for indirect fire cannot be performed simultaneously with combat actions of the launch vehicle; thus, launching events in indirect fire are sequenced as part of launcher vehicle element movement and firing events by the main program. Launching activities by ground launchers are represented by subroutine LAUNCH which is described after the following description of the missile launcher fire-support element.

Missile Launcher Fire-Support Element

An overview of the missile launcher fire-support element activities simulated by MFB is provided by the schematic shown in Figure 3.2. The process starts when an indirect-fire request is received by the launcher. Note that the launcher receives the fire request after time delays have been assessed for the MISTIC Unit Fire Controller to process the fire request data. MFB takes the fire request data and records it for the launcher. If the launcher has not had indirect-fire activities suspended, then the launcher immediately tries to confirm the mission with the FO. If the net is free then the verification discussion occurs with the FO; otherwise, the launcher waits to try again to obtain the net and send the verification message. Once the verification message occurs, the FO reevaluates the target. If the FO confirms the mission, then the launcher vehicle element is free to launch the missile. If the FO deletes the mission,

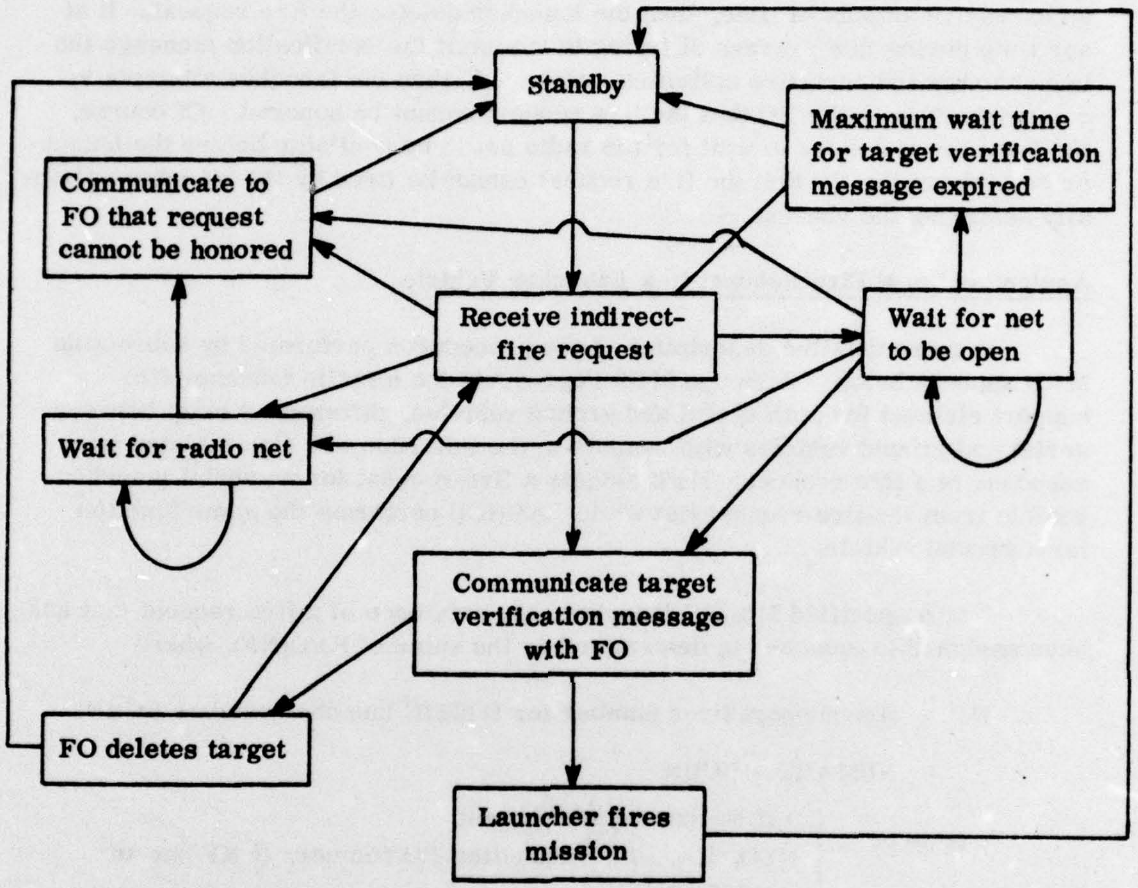


Figure 3.2.--Missile Launcher Fire Support Element

then the launcher may pick up a new fire request or go back to a standby status. If the launcher waits for the net to be open to transmit the verification message an excessive amount of time, then the launcher deletes the fire request. If at any time during this process of trying to transmit the verification message the launcher has indirect-fire activities suspended, then the launcher attempts to communicate with the FO that the fire request cannot be honored. Of course, the launcher may have to wait for the radio net to be available before the launcher can inform the FO that the fire request cannot be fired by the launcher originally accepting the request.

Assignment of a Fire Request to a Launcher Vehicle

A more detailed description of the procedures performed by subroutine MFB appears below. Although MFB represents the missile launcher fire-support element for both aerial and ground vehicles, differences exist between aerial and ground vehicles with respect to the initiation of a fire mission in response to a fire request. MFB selects a fire request for an aerial launcher vehicle from the fire request list while LAUNCH performs the same function for a ground vehicle.

For a specified MISTIC launcher, the existence of a fire request that has been assigned to launcher is determined by the value of KFO(NF), where

$$\begin{aligned} NF &= \text{fire support firer number for MISTIC launcher number LCHN} \\ &= NUMART + LCHN \end{aligned}$$

$$KFO(NF) = \begin{cases} 0 \text{ if NF has no assignment,} \\ \text{NFO, i.e., the requesting FO number, if NF has an assignment} \end{cases}$$

The value of NUMART is found in COMMON/NUMBER/. On the other hand, NFB(NFO) is fire-support firer number that is assigned to FO number NFO's fire request. For ground launchers, LAUNCH records a mission assignment for action by the missile launcher fire-support element by specifying a value for KFO(NF).

For aerial launchers, MFB attempts to search the list of fire requests to find the earliest unassigned fire request for assignment to LCHN. This search is only made if the indirect-fire activities for NF have not been suspended. Suspension of indirect-fire activities may occur when NF has to conduct a direct-fire self defense mission or when the launcher suffers a firepower or total kill. A flag is used to indicate suspension of indirect-fire activities; i.e.,

$$IFRFL(KSUB) = \begin{cases} 1 \text{ if indirect-fire activities for LCHN have been expended,} \\ 0 \text{ if otherwise} \end{cases}$$

KSUB = LCHN + ITOTFO, and

ITOTFO is found in COMMON/NUMBER/.

Having determined that the aerial launcher has not suspended indirect-fire activities, a search of the fire request list is initiated. The fire requests are subscripted by N, the fire request entry number, and NT, the fire-support unit number. NT is equal to NUMART plus the missile unit number, NMISUN(LCHN). The FO initiating a fire request is recorded by NFOFR(N, NT); thus, the existence of a launcher assigned to a fire request entry can be determined by

1. Finding the requesting FO; i.e., NFO = NFOFR(N, NT); and

2. Determining whether a launcher has been assigned which is true if NFB(NFO) > 0.

To find the earliest available fire request, the array STRTIM is used, where

STRTIM(N, NT) = firing data availability time for fire request (N, NT) from the MISTIC Unit Fire Controller.

Once MFB finds the earliest available fire request for an aerial vehicle, the assignment is recorded by

KFO(NF) = NFOFR(NSVE, NT)

NFB(NFO) = NF,

where NSVE is the earliest available fire request entry number for fire-support unit NT.

The data availability time STRTIM(NSVE, NT) may occur at some point beyond the current clock time for the missile launcher fire-support unit. Letting

NFBCLK = clock number for missile launcher fire-support element NF

TC(NFBCLK) = clock time for NF

then NF's event is terminated and the next event occurs at time STRTIM(NSVE, NT) if STRTIM(NSVE, NT) > TC(NFBCLK). Otherwise, the current event for NF may continue so that a verification message may be initiated. Of course, suspension of indirect-fire activities may occur in the event NF must wait before receiving the fire request data.

Receipt of Fire Request Data and Mission Initialization

After assignment, the launcher receives fire request data transmitted by the FO, and the recording of these data is performed by MFB. These data are simply transferred from the fire request list to the launcher firing list. The data received by the launcher are:

[XFRL(N, NT), YFRL(N, NT)] = battlefield coordinates of the target or target complex to be engaged by the launcher

NRNDFR(N, NT) = requested number of missiles

IPRIRR(N, NT) = mission priority

$$\text{MISFRL(N, NT)} = \begin{cases} 2 \text{ if launcher is to verify this mission with the FO} \\ 1 \text{ if otherwise} \end{cases}$$

For this launcher, the above fire request data are recorded in the following variables of the launcher firing list subscripted by the launcher fire-support firer number, NF:

[XD(NF), YD(NF)] = [XFRL(N, NT), YFRL(N, NT)]

NVOLM(NF) = NRNDFR(N, NT)

KPRIOR(NF) = IPRIRR(N, NT)

IFBMIS(NF) = MISFRL(N, NT)

Following transfer of the above data, the entry (N, NT) is removed from the fire request list.

In addition, several flags require initialization to control the fire mission processing. These flags are:

$$\text{IFRND(LCHN)} = \begin{cases} 1 \text{ if the launcher is set up for this mission} \\ 0 \text{ if otherwise} \end{cases}$$

$$\text{JFRND(LCHN)} = \begin{cases} 1 \text{ if the launcher has actually launched the first round of the fire request} \\ 0 \text{ if otherwise} \end{cases}$$

$$\text{MREADY(LCHN)} = \begin{cases} 1 \text{ if sufficient missiles to fire the fire request have been loaded} \\ 0 \text{ if otherwise} \end{cases}$$

The above flags are all initialized to zero.

Initiation of Target Verification Message

The purpose of the verification message is for the launcher to communicate with the FO prior to launch to verify that the FO is still alive and that the target exists. This communication with the FO will require a delay between the time the launch routine establishes the mission and the receipt of the verification. The length of this delay will depend upon whether or not the fire-request radio net is busy. Regardless of the net conditions, any delay will require that the fire-request processing procedures be suspended and that the current missile launcher fire-support element event be terminated until the delay has passed. Following the delay for this communication, MFB is again entered with all missile launching parameters set and a communication event pending. An important flag to indicate the state of a mission for the launcher upon entering MFB is:

$$\text{IFBMIS(NF)} = \begin{cases} 0 \text{ if no mission in process for firer NF,} \\ 1 \text{ if ready to launch,} \\ 2 \text{ if awaiting verification, and} \\ 3 \text{ if net busy too long, cancel mission.} \end{cases}$$

Thus, if IFBMIS(NF) is greater than zero, then the process switches to a continuation procedure. In particular, IFBMIS(NF) will be set to one by subroutine AFO once the verification message has been transmitted. A value of three will cause the mission to be cancelled because the launcher has been waiting an excessive amount of time for the net to become available. Note that IFBMIS(NF) is initialized for a new mission by the fire request data received; thus, a tactic of deleting the fire-request verification message requirement may be represented by initializing IFBMIS(NF) to one.

A description of the procedures required to initiate a verification message appears below. The first step in the target verification process is to assess whether the communication net is busy, or if a verification request can be sent. If the net is not busy, (IFDCNT(NT) = 0), then the net flag that shows the net status, IFDCNT(NT), is set to three and a request is sent. The time required for transmission over radio net NT is represented as a constant and is stored in STRMIS(NT). The radio net clock TC(NRTCLK) is updated by this amount. If the FO's activities have not been suspended (IFRFL[KFO(NF)] ≠ 1), then the FO's clock is updated by a Monte Carlo procedure giving the time for the FO to complete the verification after receiving the message. This time

is represented as a normally distributed random variable having mean USEN(NT) and standard deviation SIGSEN(NT). Finally, a maximum wait time, WAITAD(LWC), is used for the time to NF's next event and IFBMIS(NF) is set to three. If the routine MFB is reentered at time WAITAD(LWC) + TC(NFBCLK) and it is found that IFBMIS(NF) is still equal to three, this will indicate that the FO has not responded in an appropriate time and the mission will be cancelled. When the FO receives the request, he will verify it by setting IFBMIS(NF) to one and TC(NFBCLK) to the then current time. If the FO decides to cancel the mission, he leaves IFBMIS(NF) = 3 and sets NVOLM(NF) to zero. These FO procedures are described in Chapter 2. Following the setting of the times and flag IFBMIS(NF), processing by MFB is terminated until verification is received or the maximum wait time expires.

The communication net may be busy (IFDCNT(NT) $\neq$ 0) when a verification message is attempted. When this occurs, IFBMIS(NF) remains at two and the fire-support element NF attempts to transmit the verification message again after a delay of WAITFO(NT). Thus, processing by subroutine MFB is terminated until a time delay of WAITFO(NT) has passed.

Mission Cancellation

The process of establishing a mission sets up various relationships between the FO and the launcher which must be reset whenever a mission cannot be completed. As described above, a mission may be cancelled because the launcher has suffered a firepower kill, because the FO has cancelled the mission, because the time for verification has expired, because the launcher's activities have been suspended, and other reasons. Mission cancellation generates a mission termination procedures, and the procedure varies depending on whether the launcher or FO concels the mission.

Expiration of the maximum verification time or suspension of launcher indirect-fire activities causes the launcher to terminate the mission. To terminate the mission the FO must be informed; thus, the communication net is checked to see if it is busy. If it is busy, the current fire-support element event is terminated, and NF tries again to transmit a termination message after a time interval of WAITFO(NT) has passed. If the net is not busy, then KFO(NF) is set to zero and KFOD(NFO) is set to zero which will indicate to the FO that the mission is cancelled and he should pick a new target. The transmission of a termination message sets IFDCNT(NT) to three to show the net is busy, and the radio net clock TC(NRTCLK) is incremented by the time it takes to transmit an end of mission message, ENDMIS(NT). If the indirect-fire activities of the launcher have been suspended, the FO clock, TC(NFOCLK), is set to a value TC(NRTCLK) + EPSILN, slightly larger than the radio net clock, the fire-support element clock is set to infinity, and processing is terminated. If the launcher can still perform indirect-fire activities, then IFBMIS(NT) is set to

zero, TC(NFBCLK) is set slightly ahead, by EPSILN, of the radio net time, and processing of fire-support element NF by MFB is terminated until TC(NFBCLK) has occurred. In addition, ISACT(ISEC) is set to one for ground launchers to force the fire controller to reevaluate the launcher's fire position and possibly permit the launcher to move.

The cancellation of a mission by the FO results in the attempt to establish a new mission. Subroutine AFO cancels a mission by setting NVOLM(NF) = 0. When this occurs, the flag IFBMIS is set to zero, ISACT(ISEC) is set to one for ground launchers, and MFB attempts to find a new mission, if one exists. A new mission is found and an assignment made by searching the fire-request for the earliest availabe fire request in the manner described above for aerial vehicles.

Mission Verification

After mission verification occurs, i.e., IFBMIS(NF) is set to one, the missile launcher fire-support element has another event to control the timing of launch procedures. The array TIFRDY is used to control the timing of indirect-fire actions in that

TIFRDY(LCHN) = time at which launch preparation by LCHN can commence.

Thus, MFB sets

TIFRDY(LCHN) = TC(NFBCLK),

when IFBMIS(NF) = 1 to prevent launching a missile before the FO has responded with a verification message and the MISTIC Unit Fire Controller has processed the new data. In addition, launching should occur as soon after mission verification as possible; thus, MFB attempts to set the clock for the launcher vehicle element to force an event at the verification data availability times. The vehicle element clock is set by

TC(ICE) = TC(NFBCLK) + EPSILN,

where ICE is the launcher vehicle element number, if the launcher is stationary and does not have a direct-fire assignment. These conditions are identified by

EVBAR(ICE) = 0
MDFAF(ICE) = 0.

Computations in MFB

The discussion above presented the rationale for the communication activities of the missile launcher fire support element. The computations in MFB to accomplish these procedures for aerial and ground launchers are summarized below.

1. Define LCHN as the launcher being processed.
2. Determine the fire-support firer number, NF, from NUMART + LCHN.
3. Determine the clock number NFBCLK of the firer from NF + NTFB.
4. Determine the fire-support unit number NT for this launcher from NUMART + NMISUN(LCHN).
5. Determine the clock number of the unit radio net NRTCLK from NT + NTFDC.
6. Determine the weapon code LWC for unit NT from LWCOD(NUMELE + NT).
7. Set up two subscripts KSUB, the launcher index, and LCOD, the type of MISTIC missile, from KSUB = ITOTFO + LCHN and LCOD = LWC - MWART.
8. Determine the vehicle number ICE for this launcher from NOBVH(KSUB).
9. Determine the section, ISEC, from LSEC(ICE); the maneuver unit, MANUN, from LMANU(ICE); the kill status, KILL, from LKILL(ICE); fire position status, IFPC, from LFPC(ICE); and current speed, SPD, from ESPD(ICE).
10. Determine the launcher coordinates XLN, YLN by a call to ELOC.
11. If this is a continuation of a fire mission, if status flag IFBMIS(NF) > 0, then go to step 21.
12. If NF has a current assignment, i. e., KFO(NF) > 0, then go to step 17.

13. If indirect-fire activities for NF have been suspended (IFRFL(KSUB) = 1) or if the launcher is a ground vehicle (LHICE(ICE) = 0), then go to step 15.

14. If fire requests are available; i.e., NFR(NT) > 0, then go to step 38 to scan them for a new mission; otherwise set TIME = EVTIM(LWC) and go to step 16.

15. To prevent fire support launcher element from becoming the current element, set TIME = FINITY - TC(NFBCLK).

16. Update output descriptors TTIME, LSTFB(NF), and CLKFB(NF), then return.

17. Determine LAST, the number of fire requests on this fire-support unit's fire request list; i.e., LAST = NFR(NT). If LAST = 0, call ERROR.

18. Examine NFOFR(N, NT) over N = 1,... LAST to determine N, the position number of the fire request initiated by KFO(NF). If no request was initiated by KFO(NF), call ERROR.

19. Set the firing data equal to that on the fire-request list.

    a. Set target coordinates (XD(NF), YD(NF)) to (XFRL(N, NT), YFRL(N, NT)).

    b. Set the number of missiles to be fired NVOLM(NF) to NRNDFR(N, NT).

    c. Set the flags IFRND(LCHN), JFRND(LCHN), and MREADY(LCHN) to zero and BOOL to .TRUE.

    d. Set the priority of this request KPRIOR(NF) to IPRIRR(N, NT).

    e. Set the launcher status flag IFBMIS(NF) to MISFRL(N, NT) to indicate whether launcher LCHN still has to communicate with forward observer for target verification.

20. Remove entry N from fire-request list and resequence the list if necessary.

21. This step also begins the reinitialization for a continued mission. First, determine the number of the forward observer, NFO, from KFO(NF).

22. Determine this forward observer's clock number NFOCLK from NFO + NTFO.

23. If the launcher's indirect-fire activities were suspended since last cycle, go to step 26.

24. If the launcher has not become an F kill since last cycle, go to step 42.

25. Set IFRFL(KSUB) to one to suspend activities.

26. Begin to cancel mission; if the radio net is not busy; i. e., IFDCNT(NT) = 0, then go to step 28.

27. Set TIME = WAITFO(NT); i. e., update TC(NFBCLK) to call again after a wait of WAITFO(NT), then go to step 16.

28. Set status flag IFBMIS(NF) to zero to indicate that launcher LCHN has no mission.

29. Set flag IFDCNT(NT) to three to indicate the radio net is busy.

30. Set flag KFOD(NFO) to zero to indicate that FO is to select a new target.

31. Disassociate launcher and FO by setting KFO(NF) and NFB(NFO) to zero.

32. If the launcher is not an aerial vehicle, set ISACT(ISEC) to one so that the firing position will be reevaluated.

33. Set TIFRDY(LCHN) to infinity.

34. Update radio net clock TC(NRTCLK) to the time a mission cancellation message over radio net NT will be received; i. e., TC(NFBCLK) + ENDMIS(NT).

35. If the forward observer's activities have been suspended; i. e., if IFRFL(NFO) = 1, then go to step 37.

36. Update the forward observer's clock TC(NFOCLK) to TC(NRTCLK) + EPSILN.

37. If the launcher activities are suspended; if IFRFL(KSUB) = 1, then go to step 15.

38. Try to establish a new mission for the launcher by choosing the unassigned entry NSVE on the fire request list with minimum firing data availability time. If the fire request list is empty or all requests are already assigned, then go to step 15.

39. If the firing data is not yet available; i.e., TMIN > TC(NFBCLK), set TIME = TMIN - TC(NFBCLK) and go to step 16.

40. Assign the fire request; set the forward observer number NFO to NFOFR(NSVE, NT).

41. Establish the launcher-FO correspondence; i.e., set KFO(NF) = NFO and NFB(NFO) = NF, then go to step 19.

42. Continue mission, if the mission is in fire-for-effect phase; i.e., if IFBMIS(NF) = 1, then to to step 59.

43. If this is the initial event for this fire request and ICE is a ground element, set ISACT(ISEC) = 1 to cause a reevaluation of the section's firing position.

44. If the launcher does not have a requirement to communicate before firing; i.e., if IFBMIS(NF) ≠ 2, go to step 54.

45. If the net is not open; i.e., if IFDCNT(NT) > 0, then go to step 27.

46. Set flag IFDCNT(NT) to three to indicate net busy.

47. Determine time delay to communicate, FOCOM, from STRMIS(NT).

48. Monte Carlo for time FOSENS for forward observer to verify target from USEN(NT) + N(0, 1) * SIGSEN(NT).

49. Update the radio net clock TC(NRTCLK) to TC(NFBCLK) + FOCOM.

50. If forward observer's activities are suspended; i.e., if IFRFL(NFO) = 1, then go to step 52.

51. Update the forward observer's clock TC(NFOCLK) to TC(NRTCLK) + FOSENS.

52. Set TIME = WAITAD(LWC), the maximum waiting time for verification.

53. Set status flag IFBMIS(NF) to three to show the launcher standing by for verification, then go to step 16.

54. Something happened in communication cycle, either the waiting time for verification has expired, or the forward observer has cancelled mission.

55. If forward observer has cancelled mission; i. e., if NVOLM(NF) = 0, then go to step 57.

56. If the radio net is busy, so that verification is unable to get through, then go to step 27.

57. Set status flag IFBMIS(NF) to zero to indicate no mission; set IFRND(LCHN) and JFRND(LCHN) to zero, also set ISACT(ISEC) to one if ICE is not an aerial vehicle.

58. If the FO cancelled the mission; i. e., if NVOLM(NF) = 0, then set KFO(NF) = 0 and go to step 38 to find a new target; otherwise, go to step 29 to abort.

59. If the mission is finished; i. e., if NVOLM(NF) = 0, then go to step 26.

60. Set TIFRDY(LCHN) = TC(NFBCLK).

61. If the launcher vehicle is stationary and not firing a direct-fire mission, then activate the launcher immediately after the verification message by setting TC(ICE) = TC(NFBCLK) + EPSILN.

62. Go to step 15.

Ground Launcher Vehicle Indirect-Fire Activities

The model implemented in subroutine LAUNCH and designed to represent indirect-fire activities performed by a ground launcher vehicle is described in this section. The various indirect-fire events for the launcher vehicle are portrayed in Figure 3.3. Note that some of these events require communication activities by the launcher fire-support element described in the previous section. The process starts by loading missiles in the free time available to the launcher when it is not firing or moving. LAUNCH periodically inspects the fire request list to determine when the launcher will pick up an indirect-fire mission. Once a mission is selected then the missile launcher fire-support element is activated as described in the previous section. MFB receives the fire request data and

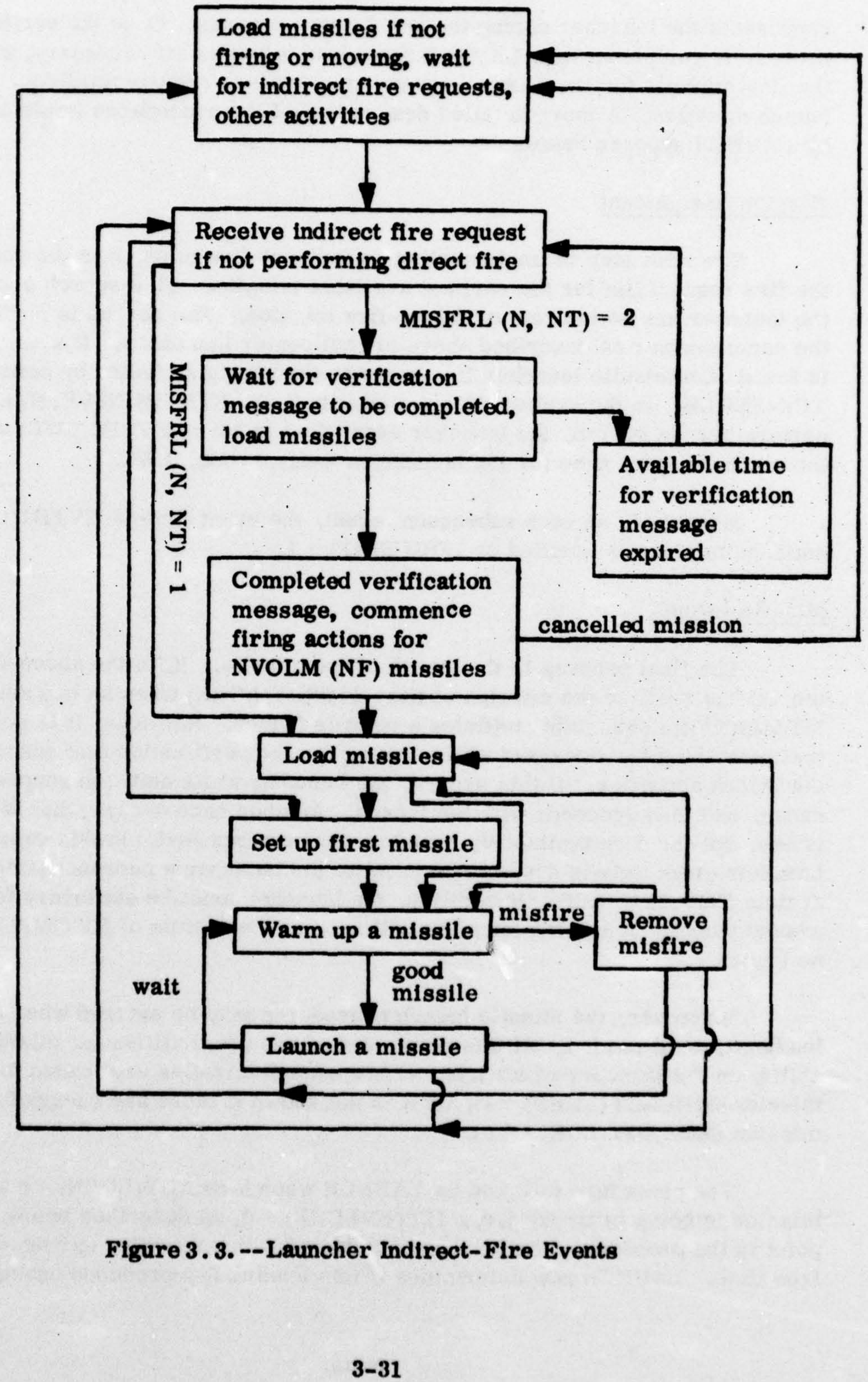


Figure 3.3.--Launcher Indirect-Fire Events

represents the launcher during the verification process. Once the verification process is complete, then LAUNCH must load missiles, if necessary, set up the first missile for the mission, warm up missiles, remove misfires, and launch missiles. A more detailed description of the procedures implemented by LAUNCH appears below.

Mission Assignment

The first step in implementing an indirect-fire mission is the search of the fire request list for the earliest available mission. This search occurs when the launcher has no indirect or direct-fire mission. The search is performed in the same manner as described above for helicopter launchers. If a new mission is found, the missile launcher fire-support element is activated by setting clock, TC(NFBCLK), to the earlier data availability time, STRTIM(NSVE, NT). After performing the search, the launcher event time is set to EVTIM(LWC) which is the standard event time for the launcher's weapon code, LWC.

Moreover, on each subsequent event, the event time is EVTIM(LWC) until the mission is verified or IFBMIS(NF) = 1.

Missile Launch

The final process is the launch of the missile. If, in the above discussion, the FO has verified the mission at time TIFRDY(LCHN) when he has set IFBMIS(NF) to one, this initiates a missile launch. However, it is possible that something has occurred while waiting for the verification that suspended the launch activities. If this occurs, the launcher waits until the suspension is ended, and then proceeds with the launch. Another case occurs when IFBMIS(NF) is one, but the data availability time has not been reached. In this case, the launcher event time is determined to make the launcher a current element again at time TIFRDY(LCHN). In addition, the launcher must be stationary for a ground launch. A moving launcher will have an event time of EVTIM(LWC) and no launch.

Moreover, the missile launch procedures may be entered when additional loading time is needed. At this point, one of two possibilities for missile availability on the launcher exist; i.e., either enough missiles are loaded for the mission (MREADY(LCHN) = 1), or it is not known if there are enough for the mission (MREADY(LCHN) = 0).

The procedure followed by LAUNCH when MREADY(LCHN) = 0 and a new mission is being initiated; i.e., IFRND(LCHN) = 0, is described below. To this point in the procedure, the launcher has been loading missiles during all of its free time. LAUNCH now determines if this loading has produced enough missiles

for the mission. Subroutine AMMO is called to determine the number of loaded missiles which is returned in the variable ICNT. If ICNT is zero and the launcher has no reserves, the mission is aborted. For a new mission, i.e., IFRND(LCHN) = 0, the launcher desires to have loaded enough missiles to fire the entire mission (NVOLM(NF)) or the maximum missile load NMLIM(LCOD); whichever is less. If ICNT ≥ NVOLM(NF) or ICNT ≥ NMLIM(LCOD), the launch procedure starts and MREADY(LCHN) is set to one. Otherwise, loading occurs until ICNT ≥ NVOLM(NF), ICNT ≥ NMLIM(LCOD), or the missile reserves are all loaded (NM(LCHN) = 0). Once one of these conditions is satisfied MREADY(LCHN) is set one. To insure that launch is initiated as soon as loading is completed, subroutine LOADM is called to load during the launcher's current event. Once loading is completed, the launcher event time is set to the load time, DELT. Also, TUSLD(LCHN) is set to -DELT to prevent loading at the beginning of the launcher's next event.

Before each missile is launched during the mission, LAUNCH verifies that at least one missile is loaded before launching. In the event that all loaded missiles are exhausted, the procedure described above to load missiles for a new mission is repeated.

After insuring an adequate number of loaded missiles, the possibility exists that set up of the loaded missile will require a setup time. The first launch for each mission assignment will be assessed a setup time which represents the slewing and laying time for the first round as a setup time, TSETUP. TSETUP is assumed to be normally distributed, and the variable TBFR(LCOD) stores the input mean value of this setup time, while SFR(LCOD) stores its input standard deviation. The flag IFRND(LCHN) is used to specify whether setup is required. If so, a value for TSETUP is determined by a Monte Carlo procedure and IFRND(LCHN) is set to one. If no setup time or no variability is desired, then TBFR(LCOD) and SFR(LCOD) or both can be input, of course, as zeros. If this is not the first launch of a mission, then the same launch position will be used and TSETUP is set to zero. The variable TIME is used to accumulate the total launch time; thus, TIME is set to TSETUP.

The selection of a missile from those loaded follows the setup exactly as in the direct-fire case. The loaded supply of missiles is decremented by decrementing the appropriate element of the LAMMO array. The launch time, TIME, is incremented by the warm up time, WT(LCOD). The possibility of a missile failure is then determined by a Monte Carlo procedure where PNG(LCOD) is the probability of a failure. If a failure occurs, TIME is incremented by TDUD(LCOD), and the current event is terminated. If no failure occurs, then the tentative launch time TM is set as TC(ICE) + TIME.

The flag JFRND(LCHN) indicates that the first round has been fired. If, at this time, JFRND(LCHN) is zero, then this will be the first round, and

JFRND(ICE) is set to one. If JFRND(ICE) is already one, then this is not the first round, and it is necessary to maintain an adequate time between each launch. The time from the last launch TD is computed from TM - EFELC(ICE) where EFELC(ICE) is the last time the launcher fired. The minimum separation time between launchings is input in SPINC(LCOD). If TD is less than SPINC(LCOD), then TM is increased to the appropriate separation by

$$TM = TM + (SPINC(LCOD) - TD).$$

The launcher will be committed until the launch, so the indirect commitment flag, TIFRDY(LCHN), is set to TM. The number of rounds to be fired, NVOLM(NF), is then decremented by one. Finally, the missile is created at time TM and given a missile number, MISNUM, by a call to CREATM(TM, MISNUM). At this point, the missile is launched. Flight of the missile is controlled by subroutine FLIGHT. The final procedure in LAUNCH is to store the launch data into COMMON/MISAVE/.

The direct-fire launch procedure assumes that the launch crew can see the target and a target number is assigned. The indirect-fire procedures assume the launch crew is firing toward a set of coordinates, and the target will be picked by the FO by use of illumination on the target upon which the missile guides, if a MISTIC, or by the controller if an E-O missile (see reference 3 for a description of these guidance procedures). Thus, the coordinates of the center of a target complex are all that are transmitted to the launcher. No target number is assigned in LAUNCH and ITARG is set to zero so that FLIGHT will determine the target illuminated by the FO.

The direct-fire situation implies direct view of the target from the launcher and exact knowledge of the target location is assumed. The indirect fire situation has uncertainty about the target location. The coordinates transmitted by the FO may be in error. If the launcher has been moving his knowledge of his exact location may also be in error. Therefore, the coordinates for the target stored in MISAVE represent the aimpoint, including location errors. The initial target aimpoint requested by the FO is stored in XD(NF), YD(NF). The standard deviation of the error for a stationary launcher is input in SDXLI, SDYLI, in COMMON/MIDATA/. The standard deviation of the error for a launcher that has moved (LIMOV(ICE)= 1) is input in SDXLM, SDYLM also in MIDATA. The actual missile aimpoint is stored in XAT, YAT after Monte Carloing for the normally distributed error about XD(NF), YD(NF) using the appropriate standard deviations. Finally, the launch angie and heading are determined by a call to LNBFLT. The completion of these data setups permits the transfer of data to MISAVE through COMMONS/LNSET/ and /OPEN/ using subroutine MICONP.

The transfer of data to FLIGHT from LAUNCH is done by subroutine MICONP, which stores data for missile MISNUM in COMMON/MISAVE/ when the data are loaded in LNSET and OPEN. The flight procedures begin by calling MICONP and replacing the data for MISNUM in LNSET and OPEN from MISAVE. The values saved by LAUNCH are listed below.

TLN	=	launch time = TM,
XLN, YLN	=	launch coordinates,
NCE	=	launcher element number = ICE,
LFO	=	FO's element number = NOBVH(NFO),
NUMM	=	launcher's launcher number = LCHN,
IFO	=	FO's FO number from NFO,
T	=	missile flight time = 0,
FINFLG	=	direct-fire flag = 0,
XAT, YAT	=	target aimpoint, and
ITARG	=	0, indicating FLIGHT picks the target.

The last process in LAUNCH is the final recording of the completion of a launch. This is done by setting the firing flag, IFIR, to two to indicate indirect fire and setting EFELCK to TM. Later in the computational procedure, the DYNTACS X main program will record the value of EFELCK in EFELC(ICE) for use in subsequent spacing of missiles in time by LAUNCH.

<u>Indirect Fire Computational Procedures in Subroutine LAUNCH</u>

Step 56 below begins the computational steps in LAUNCH for indirect fire by ground missile launchers. Steps 1 through 55 appeared earlier in this chapter, beginning on page 3-14, and were concerned with launching direct-fire missiles. After the initialization and optional loading procedures presented there, these steps follow if the launcher does not have a direct fire target assigned as a mission.

56. If the launcher has an indirect fire mission in progress (IFBMIS(NF) > 0 or |FINITY - TC(NFBCLK)| ≥ 10), then go to step 63.

57. Attempt to return launcher to indirect-fire activity by examining the fire request list for a suitable indirect-fire mission; i.e., determine the unassigned entry NSVE on the fire request list with minimum firing data availability time. If the fire request list is empty or all requests are already assigned, then go to step 61.

58. Set the FO number NFO to NFOFR(NSVE, NT).

59. Establish the launcher-FO correspondence; i.e., set KFO(NF) = NFO and NFB(NFO) = NF.

60. Reset the firing battery clock NFBCLK to the time that firing data will become available (or to CLOCK + EPSILN, if the data is presently available).

61. Set TIME = EVTIM(LWC).

62. Return.

63. If the indirect fire mission is not confirmed (IFBMIS(NF) $\neq$ 1), then go to step 61.

64. If the mission is completed (NVOLM(NF) = 0), then reset the firing battery clock NFBCLK to CLOCK + EPSILN, then go to step 61.

65. If the mission cannot be fired at this time (CLOCK < TIFRDY(LCHN)), then set TIME = TIFRDY(LCHN) - CLOCK and return.

66. If the launcher is moving (SPD > 0) then go to step 61.

67. If this is an old mission (IFRND(LCHN) > 0), then go to step 79 to check ammo supplies.

68. If the launcher has missiles loaded (MREADY(LCHN) > 0), then go to step 82.

69. Check ammo supplies for a new mission by a call to AMMO(ICE, ITYP, ICNT).

70. If the number of rounds loaded, ICNT, equals 0 and the reserves are completely depleted (NM(NUM) = 0), then abort the mission by setting NVOLM(NF) = 0 and going to step 64.

71. If there are sufficient missiles loaded to commence the new mission, then go to step 78.

72. Determine the number of rounds to be loaded, NRD, equal to the minimum of NVOLM(NF) - ICNT or NMLIM(LCOD) - ICNT where ICNT is the number of rounds already loaded. Initialize DELT to the standard event time EVTIM(LCOD) and CALL LOADM(LCHN, DELT, NRD) to attempt to load NRD rounds during this event.

73. Set TIME to DELT, the actual loading time spent.

74. Increase TIFRDY(LCHN) by DELT.

75. Set TUSLD(LCHN) to -DELT to prevent loading at the beginning of the next event.

76. If the reserves are completely exhausted; i.e., NM(LCHN) ≤ 0, then set MREADY(LCHN) = 1.

77. Return.

78. Set MREADY(LCHN) = 1 to indicate the launcher has enough missiles loaded to commence the mission, then go to step 82.

79. Check ammo supplies for a mission in progress by a call to AMMO(ICE, ITYP, ICNT).

80. If rounds are loaded; i.e., (ICNT > 0), then go to step 82.

81. Set MREADY(LCHN) = 0 to indicate launcher must reload, then to to step 70.

82. If the launcher has been set up for firing; i.e., if IFRND(LCHN) > 0, then initialize time to set up, TSETUP, to zero and go to step 85.

83. Monte Carlo for time, TSETUP, to set up launcher from TBFR(LCOD) + N(0, 1) * SFR(LCOD).

84. Show launcher ready to start, set IFRND(LCHN) to one.

85. Initialize event time TIME to TSETUP.

86. Set LNCH = 1.

87. Select a missile and decrement supply; i.e., subtract one from LAMMO(ITYP, ICE).

88. Increment event time TIME by the missile warm up time WT(LCOD).

89. Monte Carlo for missile failure (R = U(0, 1) > PNG(LCOD)) and if no failure, go to step 91.

90. Increment event time, TIME, by the reaction time for a misfire, TDUD(LCOD), then return.

91. Determine a tentative value for the launch time, TM, from TC(ICE) + TIME.

92. If the first missile has not been launched; i.e., if JFRND(LCHN) = 0, set JFRND(LCHN) to one and go to step 96.

93. Compute the time, TD, since last launch from TM - EFELC(ICE).

94. If sufficient time has elapsed since last round; i.e., if TD ≥ SPINC(LCOD), then go to step 96.

95. Adjust tentative value for missile launch time, TM, to new firing time of EFELC(ICE) + SPINC(LCOD).

96. If a missile with EO sensor is being fired (i.e., if TYPMIS(LCHN) is even), update launcher availability time, TDFRDY(LCHN), to TM.

97. Set TIME = TM - CLOCK.

98. Create a missile element numbered MISNUM by call to CREATM(TM, MISNUM).

99. Set forward observer number NFO to KFO(NF), then record launch data into COMMON/LNSET/ for the launch event by setting:
    a. launcher position (XLN, YLN, ZLN) to the position of launcher element ICE, (XE, YE, ELVATE(XE, YE, ICE));
    b. launch time TLN to TM;
    c. launcher element number NCE to ICE;
    d. forward observer's element number, LFO, to NOBVH(NFO);
    e. forward observer's FO number IFO to NFO;
    f. missile flight time T to zero;
    g. direct-fire flag FINFLG to zero;
    h. launcher's launcher number NUMM to LCHN; and
    i. target element number ITARG to zero.

100. Decrement the rounds remaining to be fired in this mission, NVOLM(NF), by one.

101. Set standard deviation of launcher location error (SDX, SDY) to the moving error (SDXLM, SDYLM).

102. If the launcher moved from initial position; i.e., if LIMOV(ICE) = 1, then go to step 104.

103. Reset standard deviation of launcher location error (SDX, SDY) to the stationary state error (SDXLI, SDYLI).

104. Monte Carlo for launch aimpoint (XAT, YAT) from (XD(NF) + N(0, 1) * SDX, YD(NF) + N(0, 1) * SDY).

105. Determine initial heading and launch angle by call to LNBFLT.

106. Record missile flight data in COMMON/MISAVE/ by call to MICONP.

107. Record data in COMMON/ICECOM/ for this event; i.e., set IFIR = 2, EFELCK = TM, LFELTK = 0.

108. The computations are complete.

VARIABLE DEFINITION INDEX

Variable	Definition	Variable	Definition
EFELC(ICE)	p. 3-34, 35	NFOFR(N, NT)	p. 3-21
EFELCK	p. 3-35	NFB(NFO)	p. 3-21
ENDMIS(NT)	p. 3-24	NFBCLK	p. 3-21, 24, 25, 32
EVBAR(ICE)	p. 3-7, 25	NFOCLK	p. 3-24
EVTIM(LWC)	p. 3-32	NM(LCHN)	p. 3-5, 12, 33
ICECOM	p. 3-14	NMISUN(LCHN)	p. 3-9, 21
ICE	p. 3-9	NMLIM(LCOD)	p. 3-6, 7, 33
IFRFL	p. 3-20, 23	NRL(LCOD)	p. 3-5, 6, 12
IFBMIS(NF)	p. 3-22, 23, 24, 25, 32	NRNDFR(N, NT)	p. 3-22
IFRND (LCHN)	p. 3-22, 5-32, 5-33	NRTCLK	p. 3-23, 24
IFDCNT(NT)	p. 3-23, 24	NUMART	p. 3-20
IFIR	p. 3-35	NUMELE	p. 3-11
IPRIRR(N, NT)	p. 3-22	NVOLM(NF)	p. 3-22, 24, 25, 33, 34
ISACT(ISEC)	p. 3-25	OPEN	p. 3-13, 34
ITOTFO	p. 3-21	PNG(LCOD)	p. 3-12, 33
JFRND(LCHN)	p. 3-22, 33	SDL(LCOD)	p. 3-6
KFIREV	p. 3-9	SFR(LCOD)	p. 3-12, 33
KFO(NF)	p. 3-20, 21, 23, 24	SIGSEN(NT)	p. 3-24
KFOD(NFO)	p. 3-24	SPINC(LCOD)	p. 3-34
KPRIOR(NF)	p. 3-22	SSR(LCOD)	p. 3-12
KSUB	p. 3-21	STRMIS(NT)	p. 3-23
LAMMO	p. 3-5, 6, 12, 33	STRTIM(N, NT)	p. 3-21, 32
LCHN	p. 3-9	TBFR(LCOD)	p. 3-12, 33
LCOD	p. 3-11, 34	TBL(LCOD)	p. 3-6
LFIRE(ICE)	p. 3-7	TBSR(LCOD)	p. 3-12
LFLAG(LCHN)	p. 3-7, 11, 13	TDFRDY(LCHN)	p. 3-11, 13, 14
LFRND(ICE)	p. 3-12	TDUD(LCOD)	p. 3-12, 33
LIMOV(ICE)	p. 3-34	TIFRDY(LCHN)	p. 3-25, 32, 34
LNCH	p. 3-9	TLOAD(LCHN)	p. 3-6
LNSET	p. 3-13, 5-34	TSETUP	p. 3-33
LRNDC(ICE)	p. 3-13	TUSLD(LCHN)	p. 3-8, 33
LWC	p. 3-11	USEN(NT)	p. 3-24
LWCOD	p. 3-11	WAITAD(LWC)	p. 3-24
MDFAF(ICE)	p. 3-25	WAITFO(NT)	p. 3-24
MIDATA	p. 3-34	WT(LCOD)	p. 3-12, 33
MISFRL(N, NT)	p. 3-22	XD(NF)	p. 3-22, 34
MISAVE	p. 3-13, 34	XFRL(N, NT)	p. 3-22
MREADY(LCHN)	p. 3-23, 32, 33	YD(NF)	p. 3-22, 34
MWART	p. 3-11	YFRL(N, NT)	p. 3-22

REFERENCES

1. Hutcherson, D. C., "Fire Support Target Selection and Coordination," Chapter 2 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

2. Hutcherson, D. C., "Fire Request Radio Net Communications," Chapter 3 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

3. Clark, G. M., et al., The Land Combat Model, Classified Annex, RF 2376 FR 71-5 (S), Systems Research Group, The Ohio State University, January 1971, AD 517621.

4. Rheinfrank, J. J. III, "DYNTACS Detection Models and the Air Defense Role of Multiple Observers," Chapter 2 (in) Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, edited by G. M. Clark et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

5. Bishop, A. B. and G. M. Clark, The Tank Weapon System, RF 573 FR 69-2B (U), Systems Research Group, The Ohio State University, Columbus, Ohio, September 1969, AD 864920.

CHAPTER 4

AERIAL UNIT MISSION CONTROL

by

D. C. Hutcherson

Introduction

In this chapter, we discuss the DYNCOM model that exercises control over the mission activities of all helicopter units. Here, decisions are made to answer the following types of questions:

Should the unit retire?

Should the unit seek a defensive position?

Should the unit discontinue its present mission and stand by for another mission?

Should the unit abort its present mission and commence a self-defense mission?

Should the unit move from its present indirect-fire loiter station to a new loiter station?

Should the unit accept a new mission from fire requests addressed to the unit?

In the model, not only are the above decisions formulated, but all communications activities are simulated that are required during negotiation of a new mission or cancellation of an old mission. Thus, it is easy to see that the model to be discussed is one of the most important models within TAPCOM II.

It should be noted that the decisions made apply to an entire aerial maneuver unit as opposed to only an aerial section. These decisions take precedence over section decisions in that they provide a frame of reference to be used in structuring section decisions. It should also be noted that the model only represents the formulation of a decision. Implementation of the decision is handled on a section basis in the Movement Control Model reported in Chapter 6. However, the Movement Controller is so designed that the sections do make movement decisions that are consistent with the unit decision.

Therefore, the Mission Control Model (MCM) of this chapter does provide the frame of reference mentioned previously.

To implement the MCM, the concept of a separate fire-support decision element is used. This same concept has been employed to represent fire-support activities of MISTIC launchers and forward observer teams (see Chapter 3 and reference 1). However, the aerial unit decision element is a completely separate entity and is associated in no way with any of the vehicles comprising the unit.

The advantage of using a separate element to represent the decision activities of aerial units is that a less complex model is permitted. The decision to commence or terminate a mission can be made by the decision element, and this element can perform any communications activities that are required because of the decision. The pace of operations of this element is determined by the rate at which decisions can be made or by the timing of the communications activities. Meanwhile, elements within the unit can proceed with movement and firing activities structured to be consistent with the mission decision that has been made. These elements do not need to be at all concerned with the communications activity--only with the mission decision. For example, if an aerial unit should decide to terminate a target-of-opportunity mission being conducted for a forward observer team (FO), the FO is notified of the decision. However, the radio net may be busy at the time that the decision is made. Therefore, the decision element must continue attempting to communicate. Meanwhile, elements within the maneuver unit can begin performing activities commensurate with the fact that the mission has been cancelled.

To implement the MCM, subroutine AIRFB has been developed and is reported in flow chart form in Volume 2. As discussed in Chapter 1, this subroutine is called from the DYNCOM main program when the aerial unit decision element NAT becomes the current element; and it is the only subroutine that processes NAT.

The reader will recall that NAT has the fire-support firer number NF = NAT + NUMART + ITOTLN and clock number NFBCLK = NF + NTFB. Both these variables were first defined in Table 1.2. Also, NAT has the fire-support weapon code LWC = LWCOD(NUMELE + NUMART + MISTUN + NAT).

The DYNCOM maneuver unit with which decision element NAT is associated is defined by the relation MANUN = KMANU(NAT), and if MANUN is known, NAT can be determined from the relation NAT = MANHEL(MANUN). It should be noted that NAT is often referred to as the aerial unit number while MANUN is referred to as the unit number. It is hoped that this abbreviated terminology will not lead to confusion.

Finally, communications activities of NAT transpire over the ground-to-air radio net originally defined in Table 1.2. This net has the number NT = NUMART + MISTUN + KP1 where

$$KP1 = \begin{cases} 5 \text{ if NAT is blue, and} \\ 6 \text{ if NAT is red.} \end{cases}$$

The clock number of this net is NRTCLK where NRTCLK = NT + NTFDC.

All of the defining constants mentioned but not defined in the preceding discussion are contained in the initialized common areas /NTELE/ and /NUMBER/. The arrays KMANU, LWCOD, and MANHEL appear in initialized common areas that bear the respective array names.[1]

Aerial Decision Element Activities

As stated in the introduction, the primary activity of element NAT is decision making. However, the rate at which these decisions may be made is controlled by the rate at which the aerial unit responds to the decisions and by the communications activities that must be performed. Moreover, the types of decisions that may be made are functions of the present activity being performed by the aerial unit and the state of the battle. We will discuss both these concepts as we give an overview of the model in the paragraphs which follow.

Model Overview

A major assumption of the model is that an aerial unit can be executing one mission while negotiating another. That is, if aerial unit NAT is executing a mission and a new fire request is received from a forward observer, NAT is allowed to request verification of the target from the FO while still executing the present mission. However, NAT does not cancel the present mission in such situations until the new mission is assigned. Moreover, the new mission is not actually assigned until a positive verification is received. Thus, the sequence of activity would be:

---

[1]Throughout the remainder of this chapter, any array discussed will be contained in a common area having the same name as the array unless otherwise noted.

1. Request verification from the FO,

2. Wait for a response,

3. Receive confirmation and assign the new mission,

4. Transmit a cancellation message to the fire-support element directing the old mission (if required).

There are several things which may be said about the procedure above, and we will discuss the procedure step by step.

Step 1

It is assumed in the model that the only types of missions requiring verification are those that are requested by forward observers; i.e., target-of-opportunity missions. In these cases, the request for confirmation triggers a response by the forward observer model reported in Chapter 1.

The model assumes that other types of missions; e.g., those requested by the battlefield command team or by the artillery intelligence center (see reference 1) do not require confirmation. Thus, the mission can be considered confirmed and the procedure can start with step 3.

Other types of missions that do not require confirmation are those that the unit decision element decides to perform without request. These missions are:

To retire.

To seek a defensive position.

To terminate the present mission and await a new request for fire.

To commence a self-defense mission.

Step 2

The wait for a response from the FO consumes an indeterminate amount of time. If the FO has not become engaged with direct-fire activities (which limit his indirect-fire activities), if he has not become a casualty, and if the radio net is open for communication when he decides to transmit, the response time may be small. However, if any one of the three inhibiting situations exist, the response time may be quite large.

The model assumes that NAT will wait only so long for a response before assuming no response is to be received. In this case, step 3 of the procedure is aborted and at step 4, the FO is notified of the cancellation. The present mission continues.

During the wait time, not all of the other decision activities are suspended. Therefore, if NAT is waiting for a response, the following decisions can be made:

To retire.

To seek a defensive position.

To terminate the present mission and stand by.

To commence a self-defense mission.

The result of this procedure is that the mission activity being performed when the response is received at step 3 is not necessarily the same as it was when the confirmation request was transmitted. Step 3 discussed below is designed to handle this possibility.

Step 3

The message received from the FO is not necessarily a confirmation message. It is possible that the FO has reevaluated the target and has decided that it no longer warrants fire from an aerial team. This can occur, for example, if elements in the target complex have moved out of the target area or if other fires have inflicted significant casualties among the target elements.

Even though a positive response is received from the FO, the new mission is not necessarily assigned. This is so because of the decisions that may have been made during the wait for a response as outlined at step 2 above. It may be that the present activity is now more important than the target of opportunity mission. In this case, the FO is notified of the rejection and the present activity is continued. Of course, if the FO has transmitted a cancellation, no message is required.

Step 4

If a new mission is assigned, the element directing the old mission is notified of the termination. Of course, it is possible that no such element exists because of the nature of the old mission. The aerial unit may have been performing one of the activities:

Waiting for a mission while off the battlefield.

Waiting for a mission while on the battlefield.

Conducting a self-defense mission.

Occupying a defensive position.

The directing element is NAT in these cases.

As outlined in steps 2 and 3 above, the element being notified of termination may be the FO. In this case, the element is not the directing element but simply an element whose mission is being rejected.

A variation of the procedure outlined above is triggered when a fire request is received from an FO and the mission is found to be less important than the present activity. This case is similar to the case in which mission decisions are made while waiting for an FO response. Instead of requesting verification from the FO, he is notified that his mission has been rejected and the present activity is continued.

The procedure outlined above is triggered any time a mission decision is being contemplated by NAT. From the discussion, it is obvious that these decisions are either concerned with one of the self-directed missions or a mission that has been requested by some other fire-support element. A definite hierarchy is assumed among these decisions; and also, assumptions are made that limit the rate at which decisions can be made. We will outline the rate limitations first.

Decision Rates

It is assumed that no decisions are allowed so long as NAT is attempting to communicate with an element for the purpose of cancellation. As outlined in the procedure discussed above, this element can be the one that was directing a previous mission or it can be an FO whose mission is being rejected. In either case, decision activities are suspended until the radio net becomes free of traffic so that the message can be sent. Also, a result is that new decisions will tend to be avoided for a short period immediately after commencement of a new mission.

Another assumption that limits the rate at which decisions can be made is that new decisions cannot be made while a previous decision is being implemented by the aerial unit. That is, a new decision cannot be made until all section leaders within the unit have had at least one event since the last decision was made. This assumption is mechanized by recording the fact that a

decision has been made and by keeping track of the sections in the maneuver unit that have not had events in which to react to the decision. Only when all sections have had a chance to react will new decisions be allowed. Again, this assumption tends to reduce the decision rate for a short period after a new mission is assigned.

Decision Hierarchy

We have already spoken of the fact that the decision to commence a self-directed mission can take place while a new mission for an FO is being negotiated. This circumstance is indicative of the fact that the model assumes that self-directed missions take precedence over directed missions, at least insofar as these missions are considered first in the decision analysis. The ultimate test used is the mission importance as indicated by its recorded priority. We will discuss priorities further when we discuss details of the model.

The decision analysis is outlined in Table 4.1. There, the decisions available to aerial unit NAT are indicated as a function of the present state of the unit. The entries in the table indicate the order in which the criteria for specified decisions are tested. A blank indicates that the decision is not available.

The state variables and activity description abbreviations appearing in the left-most columns of the table were given in Chapter 1. The reader will recall that IUNACT(NAT) is the variable specifying the current activity of aerial unit NAT while IPHASE(NAT) specifies whether NAT is enroute to the mission objective area (1) or is already in the mission objective area (0). The activity description abbreviations can be translated as follows:

NTA waiting for a mission while over the battlefield (No Team Activity).

TOP conducting a Target of OPportunity mission requested by a forward observer.

CBA conducting a CounterBattery Attack mission.

SAD conducting a Search And Destroy mission.

SDM conducting a Self Defense Mission.

IFM conducting an Indirect Fire support MISTIC launcher mission.

Table 4.1

Decision Analysis

Present Activity			Decisions						
Abbreviation	IUNACT	IPHASE	RETIRE	DEFENSE	SELF DEFENSE	NO MISSION	DIRECTED	MOVE STATION	CONTINUE
NTA	0	0	1		2		3		4
		1	1		2		3		4
TOP	1	0	1	2		3	4		5
		1	1		2		3		4
CBA	2	0	1	2	3	4	5		6
		1	1		2		3		4
SAD	3	0	1	2		3	4		5
		1	1		2		3		4
SDM	4	0	1	2		3	4		5
IFM	5	0	1		2	3	4	5	6
		1	1		2		3		4
MFO	6	0	1		2	3	4		5
		1	1		2		3		4
CBO	7	0	1		2	3	4		5
		1	1		2		3		4
SFO	8	0	1		2	3	4		5
		1	1		2		3		4
DEF	9	0	1		2	3	4		5
		1	1		2				3
RET	10	0	1				2		3
		1							1
		2				1	2		

MFO	conducting a MISTIC Forward Observer mission.
CBO	conducting a CounterBattery Observation mission.
SFO	conducting a Special Forward Observer mission.
DEF	conducting a DEFensive operation.
RET	RETiring from the battlefield or waiting for a mission while off the battlefield.

The decision abbreviations listed at the top of the table can be translated as follows:

RETIRE	the unit should commence retirement from the battlefield.
DEFENSE	the unit should seek a defensive position.
SELF DEFENSE	the unit should commence a self-defense mission.
NO MISSION	the unit should terminate its present activity and stand by.
DIRECTED	the unit should consider a new mission from the fire request list.
MOVE STATION	the unit should move its indirect-fire MISTIC loiter station.
CONTINUE	the unit should continue its present activity.

Note that the decision analysis depicted in Table 4.1 assumes that a decision is allowed in the first place. That is, NAT is not trying to communicate a mission termination message and no mission decision is currently in the process of being implemented by sections within the aerial unit. Note also that the default decision is to continue the present activity. The only case in which this decision is not allowed arises when it is determined that the unit has just arrived in the vicinity of the battlefield. The activity it has been performing (travel to the battlefield) cannot be continued. It must either begin to wait for a mission or pick up a directed mission.

We will discuss further the assumptions used in arriving at the decision analysis structure and the decision criteria used in the next section. We will also discuss more fully some of the variables used in the model.

Decision Analysis

One may notice from Table 4.1 that the decision to retire from the battlefield is an irreversible decision. Once started, the aerial unit will continue the retirement activity until a retirement position is reached and will never perform any other battlefield activity. The retirement decision takes precedence over every other mission decision that can be made.

The reason that retirement is irreversible and is so important is that the decision is made only when the unit can no longer function as an effective force. Within the model, we take as a measure of this declining capability the ratio NRET/IMEM where NRET is the number of sections from the aerial unit that have retired individually and IMEM is the number of sections that were initially present when the unit arrived in the vicinity of the battlefield. The unit will decide to retire if

$$\frac{\text{NRET}}{\text{IMEM}} > \text{RETFCN(LCOD)}$$

where RETFCN(LCOD) is input and is the retirement decision threshold for a unit with aerial unit weapon code LCOD. The aerial unit weapon code is computed by the relation LCOD = LWC - MAXLWC where LWC is the fire-support unit weapon code and MAXLWC is as defined in common area /NUMBER/ (Volume 2).

The number NRET is based upon the fact that an individual section can retire by itself as reported in Chapter 1 and Chapter 6. Criteria for this decision are:

1. The remaining fuel supply of elements in the section is low, or

2. The remaining ammunition supply of elements in the section is low, or

3. The section has absorbed significant casualties.

Both NRET and IMEM are computed in subroutine COUNT whose processing sequence is illustrated by the flow chart in Volume 2.

The decisions to seek a defensive position or to commence a self-defense mission are quite closely related, and the reader will note from Table 4.1 that the unit will always consider at least one of these decisions unless it is retiring or is off the battlefield. The reason for this is that both activities are related to the unit's assessment of the current threat posed to the unit by the enemy. The self-defense mission will be selected if the unit decides that the threat should be engaged, while the defensive operation will commence if the unit decides that the threat is too great to be engaged.

To resolve the two decisions being considered, we employ some of the basic assumptions of TAPCOM II. These assumptions are as follows:

1. While a unit is enroute to its mission objective area, sections of the unit are not allowed to attack targets. Being enroute is indicated by IPHASE(NAT) = 1.

2. While in the mission operations area, sections within the unit are allowed to attack targets only if an attack type mission is being conducted. This is indicated by MCLASS(NAT) = 2, as discussed in Chapter 1.

3. Individual sections within the unit that are attempting to select targets for attack have four decisions available to them (see Chapter 6). These decisions are:

   a. select and engage a target,
   b. continue searching for a target,
   c. retire independently,
   d. seek a defensive position independently.

As modeled, the decision for a unit to seek a defensive position is based upon the number of sections within the unit that have already decided to seek a defensive position independently. That is, the unit will seek a defensive position if

$$\frac{\mathrm{NDEF}}{\mathrm{IMEM - NRET}} > \mathrm{DEFFCN(LCOD)}$$

where

NDEF = the number of sections that are independently operating defensively

IMEM = the number of sections originally in the unit

NRET = the number of sections that have retired independently

DEFFCN(LCOD) = the input retirement decision threshold for a unit with aerial weapon code LCOD.

The variables IMEM, NRET and LCOD were defined previously in our discussion of the retirement decision. The variable NDEF is computed in subroutine COUNT whose processing sequence is given in the flow chart in Volume 2.

The ratio NDEF/(IMEM - NRET) measures the fraction of the currently available unit strength that has been lost due to defensive operations decisions within the unit. It is assumed that the higher this ratio is, the less capable the unit is to continue the present mission. When the threshold is reached, it is assumed that the unit would be better off to seek a defensive position and then possibly commence a different mission later. Defensive decisions for the sections within the unit will be discussed extensively in Chapter 6.

From the above discussion, we see that the defensive operations decision is available to a unit only if the unit is in the operations area of an attack type mission. If a unit does not qualify under these conditions, it cannot elect to operate defensively. However, such a unit can elect to commence a self-defense mission. If this is done, then the unit could subsequently elect to operate defensively since it would then be conducting an attack type mission. Furthermore, it would immediately be in its operations area by definition (see Chapter 1).

To decide to commence a self-defense mission, subroutine CMMIS is used, and we will describe this subroutine shortly. First, however, let us note that the self-defense decision is always considered when the defensive operations decision is not available as discussed above. Furthermore, there is one case where both decisions are considered. If the unit is conducting a counterbattery attack mission, the defensive operations decision is first considered and then the self-defense decision is considered. The reasoning for this convention is as follows:

> In the fire controller of Chapter 6, sections within a unit conducting a CBA mission are allowed the same four decisions discussed in a previous paragraph. However, the sections are constrained to select only artillery battery targets. The section may elect to seek a defensive position, retire or attack, but other targets in the area are not considered, thus, there may be and unobserved need to conduct a mission in which all observed targets are attackable.

Since a self-defense mission is essentially conducted against a target of opportunity, the approach used to select the mission can be similar to that used by forward observers to select targets of opportunity as discussed in reference 1. In fact, subroutine CMMIS uses the target selection procedure employed by subroutine AFO. This procedure is implemented by subroutine SELECA, which was reported in reference 2. The differences between CMMIS and AFO are that only the fire support weapon code of the aerial unit is considered when evaluating target elements, and the list of potential target elements includes all enemy elements that are known to at least one element in the aerial unit. Evaluation of a potential target element is conducted by subroutine APRIOR as reported in reference 2 while construction of the known target element list is performed by subroutine GETDET. Processing sequences for subroutines CMMIS and GETDET are presented in the flow charts which appear in Volume 2.

The decision to commence a self-defense mission is made if the priority of the best target of opportunity determined in CMMIS exceeds the priority of the mission currently being conducted. The priority is NPRIOR and is either zero (if no potential target elements exist) or is positive (if a potential target element list exists and subroutine SELECA is used). The priority of the present mission is KPRIOR(NF) and is determined at the time the mission is assigned. Here, NF is the firer number defined in the introduction.

The decision to terminate a mission and stand by is considered only when a unit is in its mission operations area. The reasoning here is that the decision is predicated upon the fact that the mission has been completed. A mission cannot be completed until the unit reaches its operations area. Of course, this decision is considered only if retirement, defense and self-defense decisions have been considered and rejected.

We will discuss the criteria for terminating missions as they are modeled in subroutine MISEND. This subroutine's flow chart appears in Volume 2.

Indirect-fire missions being conducted for the battlefield command team element are never terminated in subroutine MISEND as the model assumes that these missions are never completed. They may be terminated only if another fire request that is more important is communicated to the aerial unit (discussed in a following paragraph) or if the unit decides to commence a self-defense mission. The indirect-fire missions are the indirect-fire support MISTIC launcher mission, the MISTIC forward observer mission, and the special forward observer mission as defined in Chapter 1.

The remaining missions have one absolute requirement for termination. That is, the unit must have been in its operations area for the amount of time specified for the mission. Therefore, the required relation is

TMISUN(NAT) > DURFB(NF)

where

NF	=	fire support firer number of NAT
TMISUN(NAT)	=	time at which aerial unit NAT entered its operations area as recorded by the Movement Controller.
DURFB(NF)	=	operations time specified for the mission being flown by unit NAT.

The variable DURFB(NF) is determined at the time the mission is assigned and is either taken from fire-request data or assumes a value of zero. This topic will be discussed further in a mission assignment procedure discussion to follow.

If the mission operation time criterion is satisfied and if the mission is not an attack type mission, the mission can be terminated. Thus, counter-battery observation missions and defensive operations have only one termination criterion. However, attack missions require two further conditions for termination. First, a specified number of attacks must have been conducted by the sections within the unit. Then, if this condition is satisfied, no section within the unit can be currently conducting an attack. The first criterion is satisfied if NVOLM(NF) $\leq$ 0 where NVOLM(NF) is the number of attacks remaining to be delivered. This variable is initialized at the time of mission assignment as the number of attacks desired. Thereafter, it is decremented by one each time an attack takes place. The initial value is taken either from fire request list data or is assigned in subroutine CMMIS for self-defense missions. The initialization procedure will be discussed further in the mission assignment process description. The variable NF is the fire support firer number as discussed in the introduction.

To determine whether a given section within the unit has a target, the section movement activity indicator is used. As discussed in Chapters 1 and 6, the aerial section NSE has a target if JUNACT(NSE) = 2. To satisfy the final termination criterion, all sections in the unit must have JUNACT(NSE)$\neq$2.

We defer a discussion of the decision to consider a directed mission since it involves some fairly involved logic as discussed in the model overview given previously. Therefore, we have placed the discussion in a following subsection.

Now, we will discuss the decision to move an indirect-fire support loiter station. Referring to Table 4.1, we see that this decision is only possible when the unit is conducting an indirect-fire MISTIC mission and no other decision (including the decision to consider a directed mission) has been made previously.

The decision to move a loiter station is formulated in subroutine LOIMOV (see Volume 2). The model assumes that three conditions must be met. They are: the aerial unit must not already be moving to a station or between stations; there must exist another loiter station for the unit to occupy; and the aerial unit must be occupying a loiter station other than the one indicated by the present position of the supported MISTIC ground unit. We will discuss each of these criteria separately.

The fact that the unit is moving to a station or between stations indicates that a decision has already been made with respect to the proper position for NAT. Therefore, the decision to move a station is delayed until a station is actually occupied.

To determine whether the unit is moving between stations, the mission phase indicator defined in Chapter 1 us used. In the present case,

$$\text{IPHASE(NAT)} = \begin{cases} 0 & \text{if NAT is at a loiter station} \\ 1 & \text{if NAT is between loiter stations.} \end{cases}$$

Obviously, if IPHASE(NAT) = 1, the decision to move is rejected.

To determine whether or not the latter criteria for moving a loiter station are satisfied requires a knowledge of the mechanism used to define routes for indirect-fire MISTIC units. As discussed in a subsequent section, when an aerial unit is assigned such a mission, all the loiter stations the unit is to occupy and all the routes between loiter stations to be used are specified. The first station is located over the position the supported ground unit is presently occupying or is moving toward. Subsequent stations are located at the end points of each axis of advance listed for the ground unit which are beyond the present axis of advance. The number of routes for the aerial unit are limited to four, including the route from the present position to the first station, therefore, the number of loiter stations is also limited to four. However, it is possible that the supported unit has only one or two remaining axes of advance at the time the mission is assigned. Consequently, the aerial unit may initially have only two or three routes listed.

To determine whether or not another loiter station exists, the aerial unit must have another route listed. The conditions NPTS(K,MANUN) > 0, and K ≤ 4 are sufficient. Here, K = NAXIS(MANUN) + 1 and represents the number of the next axis of advance. Recall that MANUN is the maneuver unit corresponding to NAT. The variable NPTS(K,MANUN) represents the number of points in axis K and a zero indicates the route does not exist.

Finally, to determine whether or not NAT is occupying the proper loiter station, we employ the rule that the proper station is that which is over the end point of the axis of advance that is currently occupied by the supported unit.

The supported MISTIC unit number can be determined from KFO(NF) where NF is the fire-support firer number of NAT. The value of KFO is determined from fire-request list data at the time that the mission is assigned. To obtain the supported MISTIC unit, we use the relation

$$MUS = KFO(NF) - NUMART - ITOTFO$$

where ITOTFO and NUMART are as defined in common area NUMBER. Then, the maneuver unit corresponding to MUS is found by

$$MAN = KMANU(NUMAVT + MUS)$$

where NUMAVT is the total number of aerial units as defined in common area number.

The end point of MAN's current axis of advance is at

$$X = XAXIS(NP, MAX, MAN)$$
$$Y = YAXIS(NP, MAX, MAN)$$

where

$$NP = |NPTS(MAX,MAN)| \text{ , and}$$
$$MAX = NAXIS(MAN).$$

Then, given that NAT is on axis of advance NAX = NAXIS(MANUN), the current loiter station is at

$$XN = XAXIS(2, NAX, MANUN)$$
$$YN = YAXIS(2, NAX, MANUN).$$

Note that each aerial unit axis of advance is assumed to consist of only two points. Finally, NAT is at the proper loiter station if XN = X and YN = Y within certain machine tolerances. This concludes our discussion of the decision to move a loiter station.

All of the decisions discussed thus far are formulated by the model implemented in subroutine ATDEC whose flow chart appears in Volume 2.

Subroutine ATDEC also performs two other functions and both of these are related to units that have not yet come on to the battlefield for a mission. First of all, if IUNACT(NAT) = 10 and IPHASE(NAT) = 2, the indication is that unit NAT has just arrived in the vicinity of the battlefield and can now consider missions. Thus, the phase indicator IPHASE(NAT) is set to zero to indicate the fact that the unit is ready. No other actions are required as indicated by the entry for such units in Table 4.1. The reader will recall from Chapter 1 that the time at which this happens is specified by the input variable TIMARR(NAT).

The other processing required in ATDEC has to do with aerial units that are presently waiting for missions while off the battlefield. Since elements of the unit are not being processed by the TAPCOM II movement model (Chapter 7), fuel supplies remaining for these elements are not being adjusted to account for consumption while waiting. Subroutine FUELD is designed for this purpose and is called from subroutine ATDEC. Another purpose for which subroutine FUELD is intended is to remove sections from the unit in case these sections experience critically low fuel supplies while waiting.

The fact is that subroutine FUELD has not actually been implemented in the present version of DYNCOM. It was reasoned that the time expiring between arrival of a unit on station off the battlefield and initial mission assignment of the unit would be fairly small. Thus, the amount of fuel consumed during this interval would be small, especially if the user were to specify that such units actually are in a ready state on the ground. Thus, the errors introduced by not decrementing fuel supplies and adjusting organizations would be small.

If the user desires to implement subroutine FUELD, he can use the procedures of subroutine COUNT (Volume 2) to determine the sections in the unit. Next, the fuel remaining in each section can be decremented by the relations in Chapter 7. Then, a section with critical fuel supplies can be ascertained by the procedure of subroutine RETIRE (Volume 2 and Chapter 6). Finally, if a section is to retire because of low fuel, it can be removed from the unit formation by the procedure used in Chapter 6 when sections retire independently. We summarize the procedure for section ISE below:

1. Set JUNACT(NSE) = 4 and JPHASE(NSE) = 0 to indicate that aerial section NSE has retired (NSE = NAVSEC(ISE)).

2. Set FORMSE(ISE) = 0 to indicate that section ISE has departed the unit formation.

Directed Mission Negotiations

The decision to commence a directed mission involves some fairly involved processing so we have deferred until now the discussion of this decision. Besides, the decision is handled as part of the main processing in subroutine AIRFB and not as part of the processing of subroutine ATDEC discussed previously.

As with the decisions discussed previously, two preconditions are required before the mission can be considered. First, NAT must not be in the process of trying to communicate a mission termination message. Thus, KANCEL(NAT) must be zero. This variable is set to a positive value when the decision to send a cancellation message is made and is not reset until the message is sent. Our discussion of communications in a following section clarifies this procedure. Next, the decision to consider a directed mission is rejected if the aerial unit is still in the process of implementing a previous decision. This is indicated by the condition LMUFL(MANUN) > 0, or MANACT(MANUN) > 0.

The first variable is set in the movement controller (Chapter 6) when the leader of maneuver unit MANUN first senses that a decision has been made and is not reset there until all sections have reacted. The second variable is set when the mission is assigned (to be discussed) and is reset in the movement controller when the maneuver unit leader senses the decision.

Given that the above conditions are satisfied and that no preempting decision has been made in subroutine ATDEC (all decisions except the decision to move a loiter station), the decision to consider a directed mission can be made. The reader will recall from our model overview discussion that two states can exist at this point. First, NAT may already be negotiating a mission with a forward observer. If this is the case, we continue the negotiation process. If this is not the case, we determine whether we should attempt to begin the negotiation process. We will discuss the two cases separately.

Starting Negotiations

Starting negotiations requires that there exist a fire request that is addressed to NAT. If no request exists, NAT can only continue its present activity or possibly move its loiter station as discussed previously.

The relation MESARR(NAT) > 0 indicates that a request exists. This variable is initialized at zero at the start of a battle, it is incremented by one when a request is received by NAT (reference 3) and is decremented by one when the request is removed from the fire request list. The latter procedure will be discussed in more detail in a subsequent paragraph.

Now, even though a request exists, it may be that negotiations cannot commence. First, it is assumed that the start is preempted if the unit is presently enroute to a defensive position; i.e., IUNACT(NAT) = 9 and IPHASE(NAT) = 1. This assumption is based on the fact that we wish to delay the start of a new mission in situations in which a unit finds itself threatened. When NAT reaches its defensive position, it can commence negotiations since the threat situation will presumably have improved.

Now, given that a request exists (MESARR(NAT) > 0) and NAT is not going to a defensive position, we must determine the mission on the fire request list to be considered for negotiation. This is done by selecting the message addressed to NAT which has a minimum value for mission data availability time. That is, we take requests from the list in the order in which they become executable.

Mission data availability time is recorded as STRTIM(I,NT) for request I on fire request list NT. This variable is computed according to the procedures discussed in reference 3 and represents the battle time at which a mission is ready for execution.

To determine those requests I on fire request list NT addressed to NAT we look at the variable ITMASS(KP1,I) where

$$KP1 = \begin{cases} 1 \text{ if NAT is blue} \\ 2 \text{ if NAT is red.} \end{cases}$$

This variable is set in the requesting element operations models of reference 1 and can be translated to indicate to which aerial unit a request is directed. Therefore, in picking the mission N which has minimum STRTIM(I,NT), we analyze only those requests I such that

$$ITMASS(KP1,I) = NAT + NUMART + ITOTLN.$$

Once the request N has been selected, we must determine whether the mission data are actually available yet. Note that it is possible that the request can exist on the fire request list but

$$STRTIM(N,NT) > ECLOCK(NFBCLK)$$

where ECLOCK(NFBCLK) is the current battle time for NAT. This situation indicates that the request has been received but its information has not been processed for execution. In this case, NAT cannot commence negotiations until STRTIM(N,NT). Thus, no decision is made during this event.

Given that mission data exists for request N, it may be that the request can still not be honored. That is, it may be that the requested mission priority does not exceed the currently assigned mission priority. This situation is indicated by the relation $IPRIRR(N,NT) \leq KPRIOR(NF)$ where IPRIRR(N,NT) is the requested mission priority determined by the procedures outlined in refence 2 and KPRIOR(NF) is the assigned mission priority for NAT where $NF = NUMART + ITOTLN + NAT$. The assumption here is that missions are accepted for negotiation only if the mission, when assigned, will result in NAT performing a more important activity than it is presently performing. If the requested mission is rejected by this criterion, then NAT will notify the requesting element of the rejection instead of beginning negotiations. The request is removed from the fire request list, the positions of the other fire request list entries are adjusted to fill the vacated entry, the number of messages addressed to NAT (MESARR(NAT)) is decremented, and the element to which the rejection message is addressed is recorded. Then, NAT has a cancellation message communication event as will be described in a subsequent paragraph. As discussed previously, with NAT in this state all other decision activities are suspended until the message is transmitted.

If the mission is not cancelled as discussed above, NAT may consider the mission for assignment. However, the mission may first require verification. That is, it may be that NAT must request and receive verification from the requesting element before the mission may be assigned. From the previous model overview discussion, we recall that this procedure is required only when the request is for a target of opportunity (TOP) mission from a forward observer team. Thus, verification of request N is required only when $NFOFR(N,NT) \leq ITOTFO$ where NFOFR(N,NT) is the requesting FO number as determined by the model of FO operations reported in reference 1.

If the request does not require verification, it can be treated as a mission for which confirmation has been received. Thus, we can set $MPTR(NAT) = N$ to indicate which fire request entry has been confirmed for NAT. We can also set $MESCON(NAT) = 1$ to indicate that the request has been confirmed as opposed to being cancelled. The meaning and use of these two variables are explained further below when we discuss the processing required when NAT selects a request requiring confirmation.

If the request requires verification, NAT must commence the negotiation procedure that consists of sending a request for verification and then waiting for some response to be received from the FO. Note, however, that the procedure does not commence if NAT finds the air-to-ground radio net busy at the time the decision to negotiate is made. Rather, NAT waits for a period of time for the net to clear and then repeats the entire process of selecting the request list entry for negotiation. The reason for this procedure is that conditions may change in such a way during the wait period that some other request should be negotiated during the next event or NAT should not even conduct negotiations. The waiting procedure is discussed in more detail in our discussion of communications activities in a subsequent section.

Given that the net is open, NAT initiates the negotiation procedure. That is, NAT communicates to the FO and requests verification. Details of the communications activity are discussed in the section describing communications. However, special processing required to initiate the negotiation procedure is described below.

First of all, we require a procedure that will indicate that NAT has initiated negotiations. Also we need to be able to specify what the state of the negotiations are. This information is required during subsequent events of NAT during which negotiations are still in progress. The procedure uses two variables. First, MPTR(NAT) indicates the fire request list entry N that is being negotiated; i.e., MPTR(NAT) = N. Obviously, if MPTR(NAT) > 0 during any event, negotiations are already under way. Second, the variable MESCON(NAT) is used to indicate in what state the negotiations are. That is,

$$\text{MESCON(NAT)} = \begin{cases} 0 & \text{if NAT has not received a response from the FO} \\ 1 & \text{if NAT has received a positive response from the FO} \\ 2 & \text{if the FO has transmitted a mission cancellation message.} \end{cases}$$

Obviously, this variable must be set to zero when negotiations commence and is set to one or to two by the FO model if the FO transmits a response. The FO that is involved is NFO = NFOFR(N,NT) where N = MPTR(NAT).

Finally, the time at which the verification request was transmitted must be recorded so that during future events of NAT it can be determined whether or not NAT should continue to wait for a response. This waiting procedure is discussed in detail in the next subsection.

The time at which the verification request is sent is

$$\text{TINIT(LSUB)} = \text{ECLOCK(NFBCLK)} + \text{STRMIS(NT)}$$

where

ECLOCK(NFBCLK) = present clock time of NAT
STRMIS(NT) = time required to transmit a request for verification over radio net NT, and
LSUB = ITOTFO + NAT.

The subscript LSUB is used for NAT since TINIT is also used by forward observers as discussed in reference 1. The array STRMIS is input.

Negotiations in Progress

After negotiations have been initiated, NAT undergoes a series of negotiation waiting events that continue until one of two events transpire. Either the FO transmits a response to the request or NAT decides to terminate negotiations. We will discuss both these eventualities in the paragraphs which follow. Note, however, that the procedure has been designed so that other preempting decisions discussed previously can be made while the negotiations are under way. Thus, the status of NAT when a response is received from the FO may be different than it was when the request for verification was sent. For example, NAT can decide to retire while waiting for a response from NFO. However, no new mission negotiations can commence until the ones under way have been resolved.

The fact that negotiations are under way is indicated by the relation MPTR(NAT) > 0 as discussed previously. If this relation holds and no preempting decision is made, we first check to see if a response has been received from NFO. This is indicated by MESCON(NAT) > 0. If a message has not been received, then we determine whether or not NAT should continue to wait for a response. It is assumed that NAT will wait no longer if

1. the priority of the mission being negotiated no longer exceeds the priority of the present activity; or

2. the time expired since transmission of the request for verification has exceeded the maximum allowed wait time.

The first of these two conditions is used because it is possible that NAT has changed activities since the request was sent, as discussed previously. The condition is indicated by IPRIRR(N,NT) ≤ KPRIOR(NF) where

N = MPTR(NAT), and IPRIRR(N,NT) and KPRIOR(NF) are the priorities of the missions being negotiated and performed, respectively.

The second condition is used because it is possible that NFO can become a casualty, or his activities can be interrupted. The condition is indicated by ECLOCK(NFBCLK) - TINIT(LSUB) > WAITAD(LWC) where ECLOCK(NFBCLK) and TINIT(LSUB) are as defined previously and WAITAD(LWC) is the maximum time that a unit with weapon code LWC will wait for a response. The array WAITAD is input.

If it is decided that NAT should wait no longer, NAT attempts to transmit a mission rejection message and the processing involved is similar to that which was discussed previously when a fire request is rejected because it does not possess sufficient priority. The only difference is that we must first reset the indicators MPTR(NAT) and MESCON(NAT) to indicate that NAT is no longer negotiating a mission.

If NAT is to continue waiting for a response, the only processing required is the computation of the waiting event time. This time is input as EVTIM(LWC) where LWC is the weapon code of NAT.

If a response has been received from NFO; i.e., if MESCON(NAT) > 0, we must first determine what the response was. There are two possibilities as discussed previously. First, the FO may have reevaluated the target and decided to cancel the mission; i.e., MESCON(NAT) may be equal to two. In this case, the variables MPTR(NAT) and MESCON(NAT) are reset and NAT returns to try and find a new mission to negotiate.

In the second case, the FO may have transmitted a positive response indicating the mission is to be executed (MESCON(NAT) = 1). However, the mission may not be assigned. It may be that the requested mission priority IPRIRR(N,NT) does not now exceed the priority KPRIOR(NF) of the activity being performed. As discussed previously, this is possible since other preempting decisions may have been made during the wait for a response from the FO. If this condition holds, the mission is rejected by NAT using the procedure outlined for the case when NAT has not received a response and the request does not possess adequate priority.

Finally, if the response is received and it is positive, and if the mission request has adequate priority, the old mission is cancelled and the new mission is assigned. The assignment procedure involves processing that is outlined in the subsection below. After this assignment processing is achieved, NAT attempts to transmit a cancellation message to the element directing the old mission if there was such an element. This procedure is

discussed in the description of mission cancellation messages in a subsequent section. If no cancellation message is sent, then the event for NAT is complete. The event time in this case is EVTIM(LWC) as discussed previously.

Mission Assignment

When a movement decision has been made for aerial unit NAT in the mission control model, it is necessary to convey to the movement controller discussed in Chapter 6 the fact that a decision has been made. It is also necessary to provide data describing the new activity to be performed. These requirements arise from the fact that the decisions are made in the mission controller but are implemented in the movement controller as discussed at the beginning of this chapter.

The data that must be prepared are discussed in the following paragraphs.

MANACT(NAT).--This variable can be used to indicate that a decision has been made but has not been implemented. When a decision is made, MANACT(NAT) is set to a value that is greater by one than the value that the mission activity indicator IUNACT(NAT) (see Chapter 1) will take on when the decision is implemented. When the decision is implemented, MANACT(NAT) is reset to zero. Using this convention, the maneuver unit leader will not only know that a decision has been made (MANACT(NAT) > 0), but he will also know what the new mission is to be (IUNACT(NAT) = MANACT(NAT) - 1).

XD(NF), YD(NF).--These variables are battlefield coordinates of the new mission objective for aerial unit NAT. Here, NF is the fire-support firer number of NAT as defined at the beginning of this chapter. The position can be the center of a target complex (Missions: target of opportunity, self defense, search and destroy), the desired site of a loiter station (Missions: retirement, defense, indirect-fire MISTIC, no mission), the center of an observation area (Missions: special forward observer, MISTIC forward observer), or the reported position of an enemy battery (Missions: counterbattery attack, counterbattery observation.)

NVOLM(NF).--This variable indicates the desired number of attacks to be conducted by aerial unit NAT during the mission. If the mission is in response to a fire request, the value is prepared by the requesting element. Otherwise, the value is zero. It is one of the variables used to determine when a mission can be terminated.

KPRIOR(NF).--This variable indicates the priority of the mission to be conducted and is used to resolve conflicts between missions as indicated

previously. If the mission is in response to a fire request, the value is prepared by the requesting element. Otherwise, if the unit is retiring, its value is infinity and zero if the unit is seeking a defensive position.

KFO(NF).--In general, this variable indicates the fire-support element directing the mission. If the mission is against a target of opportunity, the forward observer team requesting fire is used. If the mission is search and destroy, indirect-fire MISTIC, MISTIC forward observer or special forward observer, the directing element is the ground-to-air communicator element. If the mission is either counterbattery attack or observation, the number of the battery against which the mission is directed is used. Finally, a value of zero is used for all other activities (defense, retirement, no mission, self defense).

DURFB(NF).--This variable indicates the time at which a mission can be terminated and is used in conjunction with NVOLM(NF). If the mission is to be conducted in response to a fire request, a value is obtained from the requesting element. No value is used when the mission is not in response to a fire request.

IFBMIS(NF).--This variable is used to indicate the status of mission negotiations in the case of a target of opportunity. However, at the time that the decision to execute the mission is made, it is set to one, no matter what type of mission is being considered.

NFB(NFO).--For target-of-opportunity missions directed by forward observer team NFO, this variable indicates the fire-support firer number executing the mission; i.e., NFB(NFO) = NF.

NAXIS(I).--This variable indicates the number of the axis of advance being occupied by maneuver unit I (I = KMANU(NAT)) at the time the decision is made. By convention, a value of zero is entered.

NPTS(N,I).--This variable indicates the number of points in axis of advance N for maneuver unit I. All valid helicopter axes of advance have two points, and for all mission activities except indirect-fire MISTIC missions, N is constrained to be one. A maximum value of four is permitted otherwise.

XAXIS(J,N,I), YAXIS(J,N,I).--These variables give the battlefield coordinate pairs for helicopter axes of advance where J = 1, 2.

For a new directed mission, all the data except those that describe the new mission's axes of advance are set within subroutine AIRFB. Data for the axes of advance are computed in subroutine STAXIS to be described in a subsequent paragraph.

The data that are set in AIRFB are taken from the information that is recorded for the fire-request list entry N associated with the new mission. That is,

XD(NF) = XFRL(N,NT),
YD(NF) = YFRL(N,NT),
NVOLM(NF) = NRNDFR(N,NT),
KPRIOR(NF) = IPRIRR(N,NT),
KFO(NF) = NFOFR(N,NT),
DURFB(NF) = DURRL(N, NT),
IFBMIS(NF) = 1,
NFB(NFO) = NF, and
MANACT(NAT) = MISFRL(N,NT) + 1.

Subroutine STAXIS described in Volume 2 operates according to the rules outlined in our discussion of the axes of advance information. That is, one axis of advance is recorded for maneuver unit I (associated with NAT) if NAT is not to perform an indirect-fire MISTIC mission (MANACT(NAT) $\neq$ 6). Otherwise, up to four axes of advance are recorded for I as will be discussed. Also, the axis of advance presently occupied by I is initialized at zero for all cases; i.e., NAXIS(I) = 0.

From a previous discussion, the reader will recall that an axis of advance J exists for maneuver unit I if NPTS(J,I) $\neq$ 0 where J is constrained to the interval J = 1, 2, 3, 4. Also, NPTS(J,I) indicates the number of points in axis J. For helicopter units NPTS(J,I) = 2 if axis J exists.

No matter what the mission, the first axis of advance exists. Therefore,

NPTS(1,I) = 2
XAXIS(1,1,I) = XLD
YAXIS(1,1,I) = YLD
XAXIS(2,1,I) = XD(NF)
YAXIS(2,1,I) = YD(NF)

where XLD, YLD are the present coordinates of the maneuver unit leader and all other variables have been discussed previously.

If NAT is performing an indirect-fire MISTIC mission, the number and location of axes of advance for I are determined from the position of the supported ground MISTIC launcher unit. From reference 2, the MISTIC unit number to be supported is MUS = KFO(NF) - NUMART - ITOTFO and the maneuver unit associated with MUS is MAN = KMANU(NUMAVT + MUS).

Also, XD(NF), YD(NF), used to define the end point of NAT's first axis of advance, is specified in reference 2 as the position of the end point of the axis of advance that MAN is presently occupying. This procedure will place NAT over MAN when NAT reaches the loiter station at the end of its first axis of advance. Thereafter, the procedure to be described in a subsequent paragraph is used to decide where NAT should move its loiter station. The procedure will continue to place NAT over MAN each time MAN executes a new axis of advance until MAN executes all its axes. However, the procedure is based upon the assumption that subsequent axes of advance for NAT coincide with those of MAN. Therefore, in STAXIS the following procedure is used to specify axes 2, 3, 4 for maneuver unit I (aerial unit NAT):

1. Determine the axis of advance currently occupied by MAN; i.e., set NAX = NAXIS(MAN).

2. Set the counter for the first axis to be assigned to I by the procedure; i.e., set K = 2.

3. If NAX ≥ 4 or if K ≥ 4, no more axes of advance can be specified for I; therefore, set

   $$NPTS(J,I) = 0 \qquad J = K,\ldots,4.$$

4. If the next axis of advance does not exist for MAN; i.e., if NPTS(NAX + 1, MAN) = 0, no more axes of advance can be specified for I; therefore, set

   $$NPTS(J,I) = 0 \qquad J = K,\ldots,4.$$

5. If the next axis does exist, specify the K^{th} axis of advance for I; i.e., set

   $$\begin{aligned} NP &= |NPTS(NAX+1, MAN)| \\ NPTS(K,I) &= 2 \\ XAXIS(1,K,I) &= XAXIS(1, NAX+1, MAN) \\ YAXIS(1,K,I) &= YAXIS(1, NAX+1, MAN) \\ XAXIS(2,K,I) &= XAXIS(NP, NAX+1, MAN) \\ YAXIS(2,K,I) &= YAXIS(NP, NAX+1, MAN). \end{aligned}$$

6. Repeat steps three through five for the remaining axes of advance for MAN; i.e., increment NAX and the axis counter K for I. Continue until steps 3 or 4 result in termination of the procedure.

Most of the data for a new mission that are not selected from the fire request list are established in subroutine PRMSET. However, the variable MANACT(NAT) is set in AIRFB. This procedure is as follows:

$$
\text{MANACT(NAT)} = \begin{cases} 1 & \text{if NAT is to terminate its present activity and stand by} \\ 5 & \text{if NAT is to commence a self-defense mission} \\ 6 & \text{if NAT is to move its indirect-fire MISTIC loiter station} \\ 10 & \text{if NAT is to commence defensive operations} \\ 11 & \text{if NAT is to retire from the battlefield.} \end{cases}
$$

Subroutine PRMSET is designed to operate according to the rules discussed previously when the new mission data were described. The only special discussion required is associated with specifying the objective position and axes of advance for NAT.

First of all, the objective position XD(NF), YD(NF), must be determined for NAT. If the decision is to seek a defensive position, to retire or to stand by, subroutine DEFPOS is used. If the decision is to move a loiter station, subroutine LOIPOS is used. Finally, if the decision is to commence a self-defense mission, the position of the objective position will have already been recorded by the model that made the decision (see subroutine CMMIS).

Subroutine LOIPOS makes use of the fact that up to four axes of advance are determined when an indirect-fire MISTIC mission is begun as discussed previously. Thus, all that must be done is to determine the position of the end point of the next axis of advance recorded for I (corresponding to NAT) and to specify this point as the objective for NAT. Thus,

XD(NF) = XAXIS(2,NAX,I)
YD(NF) = YAXIS(2,NAX,I)

where NAX = NAXIS(I) + 1.

Subroutine DEFPOS operates on the principle that positions for waiting, defense and retirement are input to the simulation. All that must be done is to select the proper position from those entered.

The input positions are specified by the coordinates

X = DELAY(KP1,1,I)
Y = DELAY(KP1,2,I)

where

$$KP1 = \begin{cases} 1 \text{ if NAT is blue} \\ 2 \text{ if NAT is red.} \end{cases}$$

The array is ordered on I as follows:

1. There are a maximum of NBDP positions input for the blue force (KP1 = 1) and a maximum of NRDP positions input for the red force (KP1 = 2). In addition, there is one position not input but reserved for the current position of the unit as will be explained below. Thus, I = 1, 2,..., max(NBDP + 1, NRDP + 1).

2. The retirement positions for the blue and red forces are recorded in the regions I = 1,...,NBRP and I = 1,...,NRRP, respectively. All these positions are input.

3. The waiting and defensive positions for the blue and red forces are recorded in the regions I = NBRP + 1,..., NBDP + 1 and I = NRRP + 1,...,NRDP + 1, respectively. All but the last entry in each case are input.

The use of the array is outlined below.

The reader will recall from previous discussions in this chapter and elsewhere that sections within unit NAT may go to waiting positions, defensive positions, and retirement positions independent of the unit. The assumption is made, however, that all sections that retire go to the same retirement position, all sections that seek defensive positions go to the same position, and so on. Moreover, if NAT makes one of the three decisions and a section within NAT has made the same decision previously, then NAT goes to the position selected by the section. Thus, subroutine DEFPOS has two special characteristics. First, it is designed to select positions for sections (Chapter 6) as well as for the unit. Second, the design is such that the position determined for the unit or for a section will be the same as the position determined for other sections if similar decisions have already been made. To accomplish this, whenever a section or the unit selects a retirement position and none has been selected before, the variable NDELPT(1,NAT) is set to ND where ND is the subscript of the position selected. If no position has been previously selected NDELPT(1,NAT) will be zero. Thus, if NAT or one of the sections desires to select a retirement position and finds NDELPT(1,NAT)

positive, then the position selected is that indicated by NDELPT(1,NAT). The same procedure is followed if a defensive or waiting position is desired but the variable NDELPT(2,NAT) is used.

Now, we must describe the procedure to be used when a position is not already recorded for the unit. The procedure differs slightly depending upon whether or not a retirement position is desired. It is assumed that a retirement position must be selected from the input positions. However, a defensive or waiting position can either be one of the input positions or it can be the current position of the unit or section selecting the position. This position (X,Y) is taken as the position of the leader of the maneuver unit (MANLDR(MANUN)) if NAT is making the decision or the leader of the section (ISORG(1,ISEC)) if section ISEC is making the decision. If this position is selected, then it is recorded in the reserved position with index NBDP + 1 or NRDP + 1. For example, if NAT is a member of the blue force, then

DELAY(1,1,NBDP + 1) = X,
DELAY(1,2,NBDP + 1) = Y, and
NDELPT(2,NAT) = NBDP + 1.

The procedure for selecting a position is illustrated by the example below. It is assumed that NAT is blue and that a defensive position is desired. It is also assumed that NDELPT(2,NAT)= 0 and that the decision is for the unit NAT as opposed to one of the sections.

1. Determine the intelligence available to the unit by using subroutine GETDET to combine the intelligence currently possessed by each element operating with the unit. This intelligence is recorded in the output array KDET with one entry for each enemy element.

2. Analyze each input defensive position DELAY(1,1,I), DELAY(1,2,I) I = NBRP + 1,...,NBDP and the position of MANLDR(MANUN). Construct a list of those positions which satisfy the following conditions:

   a. The position is outside the estimated effective range (EW(IT)) of each known (KDET(I) > 0), surviving (LKILL(I) ≤ 2) enemy weapon I with weapon code IT = LWCOD(I).

   b. The position is outside the estimated effective range of all elements suspected to be located at each enemy strong point J (J = 1,... NRSP) where strong point J

is located at S(J),T(J) and the maximum effective range of elements suspected at J is ET(NRWP) and NRSP and NRWP are contained in common area /SPTS/.

3. Select the position recorded in the list constructed in step 2 that is closest to MANLDR(MANUN). If the list is empty record the position of MANLDR(MANUN) as the best available position.

The procedure used when a retirement position is desired is different since the leader's position is excluded from analysis. Moreover, if the list constructed in step 2 is empty, the input position closest to the leader is selected. If a section decision is being made, then the leader is ISORG(1,ISEC) as opposed to MANLDR(MANUN).

With the objective chosen, subroutine PRMSET continues by defining the axis of advance information for MANUN. Now, axis of advance information is not required for units moving indirect-fire loiter stations since these axes are established by subroutine STAXIS when the mission is assigned. Moreover, the same is true for self-defense missions--the axis of advance is specified by subroutine CMMIS. Therefore, the only remaining missions requiring axis information are the ones for which subroutine DEFPOS was used to choose the objective; viz, defensive operations, retirement, and waiting operations.

The procedure is quite simple. There is only one axis of advance and it is defined as in subroutine STAXIS; i.e.,

```
NPTS(1,MANUN) = 2,
NPTS(J,MANUN) = 0          J = 2,3,4,
XAXIS(1,1,MANUN) = XLD
YAXIS(1,1,MANUN) = YLD
XAXIS(2,1,MANUN) = XD(NF)
YAXIS(2,1,MANUN) = YD(NF)
```

where XLD,YLD is the present position of MANLDR(MANUN) and XD(NF), YD(NF) is the objective chosen in the subroutine DEFPOS.

There is one final bit of processing required when a new mission is assigned if it is the first mission to be assigned to NAT. As explained in Chapter 1, the elements of NAT do not become active until a mission is assigned. Therefore, in this case, the clocks of the elements in the unit must be set so they will become current elements. This processing is accomplished in subroutine STACLK.

First, it is assumed there is a delay between the time the mission is assigned and the elements actually appear on the battlefield and commence their mission. This assumption is based on the fact that the unit is initially off the battlefield and is either setting on the ground in a ready status or is loitering at some position adjacent to the battlefield. The delay time is REACT(NAT) and is input to the simulation.

Now, the procedure used to set the clocks is designed to insure that the maneuver unit leader is the first element to become current. Then, the section leaders are next to become current and finally the other elements in the unit become current. This procedure is required because of the method of analysis used in TAPCOM II as explained in Chapter 1 and elsewhere. Thus,

ECLOCK(LDR) = ECLOCK(NFBCLK) + REACT(NAT)

where LDR = MANLDR(MANUN) and NFBCLK is the clock number of NAT. Then, ECLOCK(NSAVE) = ECLOCK(LDR) + R for each section leader NSAVE in MANUN where R is a uniformly distributed random number on the interval (0,1). Finally, ECLOCK(NELE) = ECLOCK(NSAVE) + R for each element NELE in a section with leader NSAVE.

As a final step in STACLK, the fuel of each section in MANUN is adjusted to account for the delay in coming onto the battlefield. That is,

WFUEL(NSEC) = WFUEL(NSEC)-REACT(NAT)*RFUEL(2,KCOD)

for each aerial section number NSEC in MANUN. The aerial section number of a section ISEC is NSEC = NAVSEC(ISEC) where the array NAVSEC is input. The array WFUEL is also input as the initial fuel supply for the sections when they arrive in the vicinity of the battlefield. The array RFUEL describes the rate of fuel expenditure for elements in a section and is explained in detail in Chapter 7. The variable KCOD is the weapon code of the aerial section; i.e., KCOD = LWCOD(NSAVE) - MAXLWC where NSAVE is the section leader and MAXLWC is defined in common area /NUMBER/.

Communications Activities

The reader will have noted that many events for NAT are involved with communications. Either NAT has transmitted a request for verification to a forward observer or NAT has transmitted a mission cancellation notice. In other events, NAT is standing by for the radio net to clear so that one of the above messages can be sent.

If NAT is standing by to send a message, the event time is WAITFO(KSUB) where the WAITFO array is input and

KSUB = ITOTFO + ITOTLN + NAT.

NAT will be processed again at ECLOCK(NFBCLK) + WAITFO(KSUB).

However, if NAT has transmitted a message, the event time is the message duration as explained below.

Verification Requests

When NAT desires to transmit a request for verification, the party to whom the request is addressed is NFOFR(N,NT) where N = MPTR(NAT) as explained previously. The reader will recall that this element is an FO. The communication time (event time) is STRMIS(NT) which is an input variable.

Now when the message is sent, the following processing is required.

1. The net must be occupied with traffic so that no other element can use the net until it clears. From reference 3 and elsewhere, the variable IFDCNT(NT) must be set to three.

2. The FO must be returned to action if possible. From reference 1, FO's commence waiting periods when they transmit fire requests so NAT must return the FO to action to get a response. This is accomplished by setting the FO's clock to an appropriate value; i.e., ECLOCK(NFOCLK)=ECLOCK(NFBLCK)+TIME+FOSENS. Here, NFOCLK is the clock number of the FO, NFBCLK is the clock number of NAT, TIME is the communications time discussed above, and FOSENS is the amount of time that the FO will take to reevaluate the target. This last value is computed by the relation FOSENS = USEN(NT)+N(0,1) *SIGSEN(NT) where the arrays USEN and SIGSEN are input means and standard deviations of the sensing time distributions and N(0,1) is a random number from a normal density with zero mean and unit variance. Note the FO will not become the current element until the sensing activity has already taken place.

   The above procedure is not followed if the FO's activities are suspended at the time the message is sent. Suspended

activities were discussed in reference 1 and are indicated by IFRFL(NFO) = 1 where NFO is the FO number. If this relation holds, the clock of NFO is not adjusted.

Cancellation Messages

The reader will recall that cancellation messages are almost always sent when a mission is being executed and is terminated. The exceptions occur when NAT has been in a defensive position or has been waiting for a mission. Cancellation messages are always sent when a fire request is rejected for any reason. In both cases above, the party to whom the message is addressed is recorded as KANCEL(NAT). The reader will also recall that other decision processes are suspended as long as NAT is attempting to transmit such a message; i.e., when KANCEL(NAT) > 0.

In the first case outlined above, the receiving element is KFO(NF); i.e., KANCEL(NAT) = KFO(NF). This variable is set when the mission is assigned as outlined in a previous discussion. In the second case, the receiving element is determined from the fire request list entry data; i.e., KANCEL(NAT) = NFOFR(N,NT) where N is the entry being cancelled.

The procedure used to simulate transmission of the cancellation is similar to the procedure outlined above for verification request transmission. The net is filled with traffic by setting IFDCNT(NT) = 3. The message transmission time is ENDMIS(NT) where the ENDMIS array is input. Finally, if the message is addressed to an FO, the FO is returned to action if possible. Again, the FO may be in a suspended acitivity state (IFRFL(NFO) = 1) so the procedure is not followed in this case. However, if activities are not suspended, the FO's clock is set to the time at which transmission will be complete (ECLOCK(NFBCLK) + ENDMIS(NT)). Also, KFOD(NFO) is set to zero so that NFO will have a target selection event in his next event.

Note that the message may not be addressed to an FO. This is so because several types of elements generate missions as explained in reference 1. In this case, the receiving element does not need to be returned to action. To determine whether or not the receiving element is an FO, the variable KANCEL(NAT) is used. The model is constructed as has been discussed in this chapter and elsewhere so that if KANCEL(NAT) ≤ ITOTFO, the requesting element is an FO. Otherwise, the requesting element is either the ground-to-air communicator or the artillery intelligence center. Special processing required for this last element is described below.

AIC Transactions

Any time a counterbattery mission is accepted for execution or is terminated or rejected, processing must occur to provide information for use by the counterbattery model described in reference 4. Subroutine CBCONT is designed for this purpose. Since the data were described in detail in reference 4, we will only present a summary of the processing that is actually accomplished.

If a counterbattery mission is rejected before being assigned, it is recorded as an unsuccessful mission. This is accomplished by setting IHCCOM(NBT) = 0 where NBT is the enemy battery against which the request was directed. The subscript NBT is found by the relation

$$NBT = NFOFR(N,NT) - ITOTFO - 2$$

(see reference 1). Also, if the mission was for a counterbattery attack, the variable LBTDET(NBT) is set to five to indicate that NBT was not attacked.

If a counterbattery attack mission is assigned for execution, data must be recorded to indicate that the mission is under way. This is accomplished by performing the operations

$$IHCCOM(NBT) = 0$$
$$LBTDET(NBT) = 4.$$

The first operations records the fact that the mission has not been completed. The second operation records the fact that the mission is being performed.

Finally, if a counterbattery mission is terminated after having been assigned, the following processing is performed:

If the mission was an attack mission, set LBTDET(NBT) = 5 to indicate the mission has been concluded.

If the mission was an observation mission, and it was unsuccessful, set IHCCOM(NBT) = 0 to indicate that the mission was unsuccessful. A mission is unsuccessful if NVOLM(NF) > 0 at mission termination (see Chapter 6).

If the mission was a successful observation mission, set IHCCOM(NBT) = 2 to indicate observation took place. In addition, compute the position of battery NBT that was observed. This position is XAIC(NBT), YAIC(NBT) and contains errors

that are functions of the amount of observation time that was accomplished during the mission. The computation procedure is:

1. Compute the observation time; i.e.,
   TIM = ECLOCK(NFBCLK) - TMISUN(NAT)
   (see Chapter 6 for a definition of TMISUN(NAT)).

2. Compute the standard deviation of the observed position error distribution; i.e.,
   SD = CBERR(1, LWC)*exp(-TIM*CBERR(2, LWC))
   where LWC is the weapon code of NAT and the array CBERR is input. Note that SD is an exponential function with intercept CBERR(1, LWC) and initial slope -CBERR(2, LWC)*CBERR(1, LWC).

3. Compute the errored observed position
   XAIC(NBT) = XFB(NBT) + R1*SD
   YAIC(NBT) = YFB(NBT) + R2*SD
   where XFB(NBT), YFB(NBT) is the true position of NBT and R1 and R2 are random numbers from a normal density with zero mean and unit variance.

Note that the above procedure for computing the errored position of NBT is based upon the hypothesis that errors can be reduced with increased observation time. However, the form of the standard deviation decay function is only postulated. Moreover, the distribution of errors is assumed to be normal and uncorrelated in X and Y. Also, no account is made of the fact that the errors may have higher variance in range than in deflection.

VARIABLE DEFINITION INDEX

Variable	Definition
T	p. 4-31 (input)
TIMARR	p. 4-17 (input)
TINIT	p. 4-22
TMISUN	p. 4-14
USEN	p. 4-33 (input)
WAITAD	p. 4-23 (input)
WAITFO	p. 4-33 (input)
WFUEL	p. 4-32 (input)
XAIC, YAIC	p. 4-36
XAXIS, YAXIS	p. 4-25
XD, YD	p. 4-24
XFB , YFB	p. 4-36 (input)
XFRL, YFRL	p. 4-26
XLD, YLD	p. 4-26

REFERENCES

1. Hutcherson, D. C. "Fire Support Target Selection and Coordination," Chapter 2 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

2. Hutcherson, D. C. "Target Selection Procedures for Aerial Vehicle Units," Chapter 4 (in) Small Unit Combat Simulation, Air Defense Operations Models, edited by G. M. Clark, et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

3. Hutcherson, D. C. "Fire Request Radio Net Communications," Chapter 3 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

4. Clark, G. M. and S. H. Parry, Small Unit Combat Simulation (DYNTACS X), Counterbattery Fire Models, RF 2978 FR 70-1 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1970, AD 882877.

CHAPTER 5

COMMUNICATIONS AND INTELLIGENCE FOR AERIAL ELEMENTS

by

D. C. Hutcherson

Introduction

Within DYNCOM, intelligence is defined as the degree of knowledge that a given element has with respect to each and every enemy element and with respect to the terrain over which the battle is being conducted. In this chapter, we are concerned with the former category of intelligence, since it is assumed that an aerial element has perfect knowledge of terrain characteristics that affect his operations. Such characteristics would be the macro-terrain profile, the micro-terrain surrounding an enemy element, and the vegetation covering the battlefield. Soft soil, rough terrain, minefields and other similar terrain characteristics do not affect helicopter operations.

Knowledge of the enemy is specified by constructing an array LDET for each current element processed (including helicopters). The array gives the present state of knowledge that the current element possesses with respect to each enemy element. That is

$$\text{LDET(I)} = \begin{cases} 0 & \text{enemy element I is unknown to the current element,} \\ 1 & \text{enemy element I is known to exist by the current element but is not at present visually detected,} \\ 2 & \text{enemy element I is visible and detected by the current element, and} \\ 3 & \text{enemy element I is the target of the current element, but is completely concealed or the current element is neutralized.} \end{cases}$$

For a given current element, DYNCOM assumes that the knowledge possessed at the start of the element's event is that which existed at the start of the element's previous event, modified by changes that occurred during the previous event. Thus, at the end of each event, the array LDET (which represents knowledge at the beginning of the event) is stored for reference at the beginning of the current element's next event. Then, at the start of the next event, the array LDET is modified to account for changes during the last event.

Two models are required in DYNCOM to modify the array LDET. The models are a communications model (subroutine COM) and an intelligence model (subroutine INTELL). These models process all DYNCOM vehicular elements including helicopters.

The communications model processes intelligence and tactical messages for various elements using the platoon, company, and battalion tactical nets. These messages are ordered in time so that messages attempted first are sent first, but queues are formed when a potential sender cannot access a busy net. For example, any element (including a helicopter) that detects an enemy element previously unknown to the detecting element, will generate an intelligence message. This message will be placed on a list of messages waiting to be sent, and the time that the message was generated will be recorded on this list. The communications model removes messages from this list (in the order of arrival) and makes the contents of the message known to elements on the radio net. In the event queues are formed, a sender is selected randomly when a net eventually becomes available, and then all messages desired to be transmitted by this sender are transmitted. In any case, only those messages generated prior to the battle time at which the communications model is called will be processed.

Thus, the communications model can modify the LDET array for the current element. If the message is an intelligence message about element I and the current element has no knowledge of element I (LDET(I) = 0), then LDET(I) is set to one. The current element has at least communicated knowledge of element I's existence and approximate location.

The important fact to note about the communications model is that it treats all vehicular elements including helicopters. Thus, subroutine COM (reference 1) is the first subroutine to be called within TAPCOM II for a current aerial element. For a detailed discussion of the communications model, see reference 1.

The intelligence model is the second model that may modify the LDET array of the current element. This model processes each current element (including helicopters) immediately upon completion of processing by the communications model. It is in this model that changes brought about by events other than message arrivals are treated. Such phenomena as visual acquisition by search, visual acquisition due to firing of the enemy, pinpointing, and loss of acquisition through interrupted lines of sight are modeled. For a more complete description of the intelligence model, see references 1, 2, 3, and 4, and the material to be presented below.

Intelligence Model Changes

There have been some slight modifications to the intelligence model since publication of the cited references. These modifications have been made for three reasons:

1. to make the model more flexible in describing search from an aerial vehicle,

2. to make the model more flexible in the treatment of aerial target detection by ground observers, and

3. to bring the model into agreement with other models of TAPCOM II reported elsewhere in this volume.

We discuss these changes below.

Aerial Vehicle Target Detection

The detection of aerial vehicle targets by ground elements is represented by the model reported in reference 4. In that model, detection is dependent upon several factors; the following are included among them:

1. A1, A2, A3 cross-sectional presented area of an aerial target as viewed from the front, the side and the bottom, respectively.

2. RF average surface reflectance of an aerial vehicle target.

3. SILL ambient illuminance level.

4. SL ambient luminance level.

5. VR meteorological visibility range.

The first four variables above are descriptors of the target, while the last three variables describe the medium or conditions under which search is being conducted.

In DYNCOM, it has been decided that the model should be more responsive to changes in these variables. While the last three variables (SILL, SL and VR) are considered constant, the first four (A1, A2, A3 and RF) are assumed to be characteristics that change as the type of aerial target changes. To being about this dependence, the following definitions are used.

7/

AVA(I,J) = cross-sectional presented area of an aerial vehicle of type J when viewed from aspect I (real, meters^2) where

$$I = \begin{cases} 1 \text{ indicates a front aspect} \\ 2 \text{ indicates a side aspect} \\ 3 \text{ indicates a bottom aspect, and} \end{cases}$$

J = aerial weapon code of the target LWCOD(IT) - MAXLWC, and

IT = target element number.

RF(I) = average target surface reflectance for an aerial vehicle target having weapon system code I (real, foot-lamberts/ lumen/feet^2) where

I = LWSYS(LWC),

LWC = LWCOD(IT), and

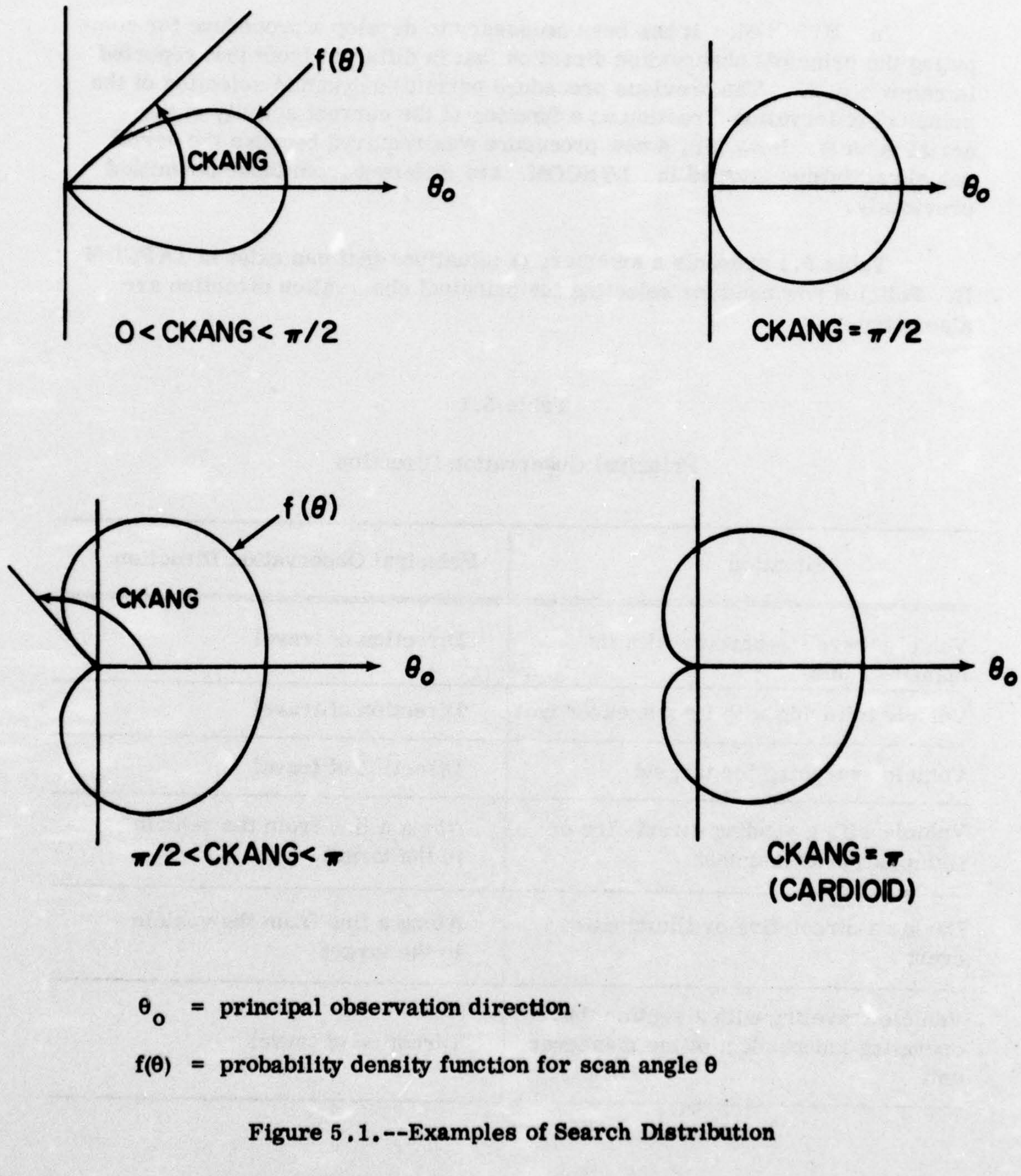
IT = target element number.

From the above definitions, we see that the presented areas are dependent upon target weapon code LWCOD while the surface reflectance is dependent upon target weapon system code LWSYS. These codes are defined for each element in common areas /LWCOD/ and /LWSYS/, respectively, while MAXLWC is defined in common area /NUMBER/. The arrays AVA and RF appear in common areas /AVA/ and /RF/ while the three variables describing meteorological conditions are contained in common area /SKY/. See Volume 2 for descriptions of each of the common areas mentioned.

The subscripting changes outlined above necessitated changes in subroutine DETH which implements the aerial vehicle detection model under discussion. The changes are only those required to obtain proper values from the new common areas discussed, so they will not be described here. However, a revised flow chart of subroutine DETH does appear in Volume 2.

Search From an Aerial Vehicle

As reported in reference 2, an aerial observer is assumed to allocate search effort in a horizontal plane according to a limacon probability density function. Examples of such a search distribution are illustrated in Figure 5.1. The reader will note that the search pattern is oriented symmetrically about



θ_o = principal observation direction

$f(\theta)$ = probability density function for scan angle θ

Figure 5.1.--Examples of Search Distribution

an axis defined as the principal observation direction and that the search pattern is completely defined by specifying the limiting scan angle CKANG.

In DYNCOM, it has been necessary to develop a procedure for computing the principal observation direction that is different from that reported in reference 2. The previous procedure permitted dynamic selection of the principal observation direction as a function of the current activity of the aerial vehicle. However, a new procedure was required because the aerial vehicle activities allowed in DYNCOM are different from those permitted previously.

Table 5.1 presents a summary of situations that can exist in TAPCOM II. Policies now used for selecting the principal observation direction are also given.

Table 5.1

Principal Observation Direction

Situation	Principal Observation Direction
Vehicle traveling enroute with its maneuver unit	Direction of travel
Vehicle loitering with its maneuver unit	Direction of travel
Vehicle searching for targets	Direction of travel
Vehicle with a pending direct-fire or illumination assignment	Along a line from the vehicle to the target
During a direct-fire or illumination event	Along a line from the vehicle to the target
Vehicle traveling with a section that is operating independent of the maneuver unit	Direction of travel

For those cases in Table 5.1 in which the direction of travel is indicated as the principal observation direction, a procedure identical to that reported in reference 3 is used to compute the angle. This procedure adjusts the observation direction to account for the observer vehicle's desired position in its section formation pattern. In summary, the procedure assumes that each section has a "center of observation" whose coordinates are specified by input data and are measured in the coordinate system used to specify the arrangement of the section formation. A line passing through the center of observation and the desired position of the observer vehicle defines the principal observation direction. In general, the observation direction for a vehicle in the section will be slightly offset from the true direction of travel.

As shown in Table 5.1, the principal observation direction for a vehicle with a target is toward the target. If the target is to be engaged or has been engaged with direct fire, the target element number is MDFAF(ICE) where ICE is the observer vehicle number. The location of this target is determined by a call to subroutine ELOC. If the assignment is for illumination, the target is located at coordinates XDFO(NUM), YDFO(NUM) where NUM is the forward observer number of ICE.

To incorporate the procedure outlined above, the principal observation direction logic for aerial vehicles has been revised in subroutine INTELL. This logic appears at the very beginning of the subroutine as shown by the flow chart of the revision that appears in Volume 2

The limiting scan angle CKANG referred to earlier appears in common area /CKANG/ (Volume 2). In the original model, one value for this variable was allowed and applied to each aerial vehicle being represented. In DYNTACS X, changes have been made to permit a more representative model. It is assumed that two different modes of search are employed by aerial vehicles. In one mode, the observer scans a wide area while in the other mode, a narrower region is of interest. The first mode would be employed in situations in which the observer was attempting to locate targets while the second mode would be used when the observer's interest was held by a target. Therefore, the variable CKANG is now input as follows

CKANG(I) = angular limit of search (real, radians)(see Figure 5.1)

where

$$I = \begin{cases} 1 \text{ for wide angle search, and} \\ 2 \text{ for narrow angle search.} \end{cases}$$

A subscript of two is used when the observer has a direct-fire target or when the observer is acting as a forward observer and has a target to be engaged within indirect fire. In all other cases, a subscript of one is used.

Finally, as reported in reference 2, it is assumed that the structure of the aerial vehicle can inhibit search. That is, it is assumed that a region to the rear of the vehicle cannot be scanned. This region is specified by input data by defining a region to the front that can be scanned. This region has an angular width of 2*ANGLIM and is oriented symmetrically about the center line of the vehicle.

In DYNCOM, the variable ANGLIM is contained in common area /ANGLIM/ and has been made sensitive to the type of aerial vehicle being analyzed (see Volume 2). The previous model assumed that all vehicles were the same. To accomodate the variety of aerial vehicles that may be represented in DYNCOM, a more flexible procedure is required. Thus, the variable ANGLIM is now input as follows

ANGLIM(LWC) = angular limit of field of view (real, radians) (see Figure 5.2)

where

LWC = aerial vehicle weapon code.

The reader will note that a different entry is allowed for each type of aerial vehicle being represented. The aerial vehicle weapon code for an element ICE that is a helicopter can be found from the relation

LWC = LWCOD(ICE) - MAXLWC.

The variable MAXLWC is found in common area /NUMBER/ while LWCOD specifies the vehicle's weapon code.

One difficulty possibly arises because of the structural interference specified by ANGLIM. It may be that the principal observation direction THETA0 specified in Table 5.1 cannot be achieved. An example of this situation is given in Figure 5.2. Here, the vehicle has a heading of DIR and the value of THETA0 desired cannot be achieved because of the right-hand cockpit visibility limitation. Within the model, a procedure is used to alleviate the difficulty. The convention is that the visibility limit closest to the desired observation angle is chosen. In Figure 5.2, the angle $\overline{\text{THETA0}}$ would be chosen.

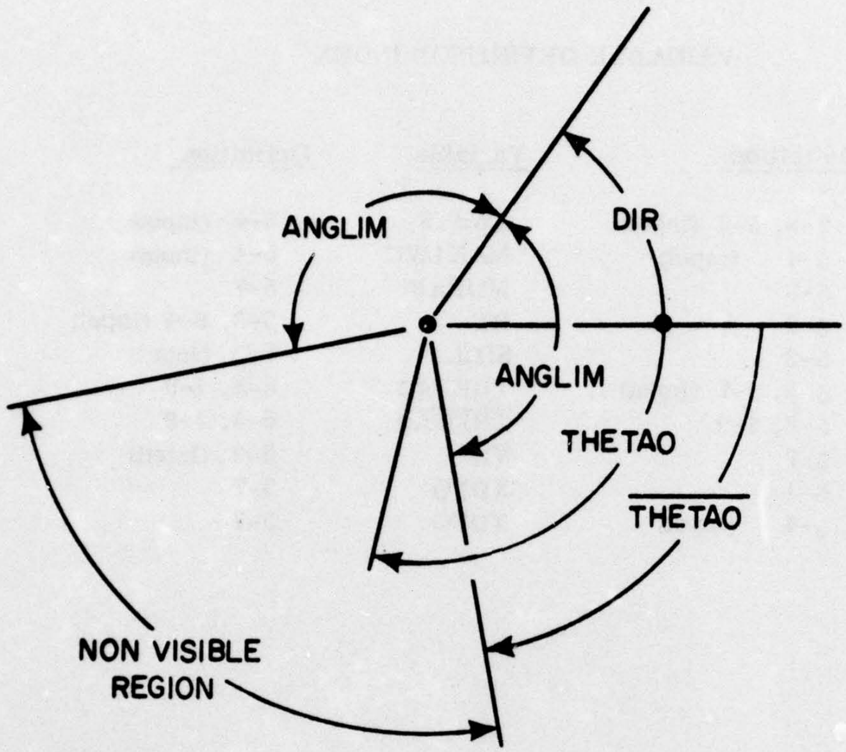


Figure 5.2.--Field of View Limitations

VARIABLE DEFINITION INDEX

REFERENCES

1. Rheinfrank, J. J. III and G. M. Clark, "Intelligence Communications," Chapter 3 (in) Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, edited by G. M. Clark, et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

2. Hutcherson, D. C., "Intelligence," Chapter 2 (in) The Land Combat Model, The Aerial Platform Combat Operations Module, edited by G. Clark and D. Hutcherson, RF 2376 FR 71-3 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, May 1971, AD 887264.

3. Clark, G. M., "Intelligence," Chapter 4 (in) The Tank Weapon System, edited by A. Bishop and G. Clark, RF 573, FR 69-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1969, AD 864919.

4. Lawson, R. M., "Air-to-Ground Detection and Ground-to-Air Visual Detection Models," Chapter 13 (in) The Land Combat Model (DYNCOM), edited by A. Bishop and G. Clark, RF 2376, FR-1 (U), Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

CHAPTER 6

AERIAL SECTION MOVEMENT AND FIRE CONTROL

by

D. C. Hutcherson[1]

Introduction

In this chapter, we discuss the DYNCOM model that exercises control over the movement and firing activities of all helicopter sections. Here, decisions are made to answer the following types of questions:

Should the section retire from the battlefield?

Should the section commence search for direct-fire targets?

Should the section select a target complex to be engaged with direct fire?

Should the section seek a defensive position?

Should the section commence a loitering operation?

Should the section commence movement toward a new mission objective?

In the model, not only are the above types of decisions formulated, but all processing required to implement the decisions is performed. Thus, routes and formations are chosen and recorded for the section, targets are selected and assigned to elements in the section, firing sequences are formulated for elements with firing assignments, and then movement is monitored to determine when new movement and firing decisions can again be made. The model is clearly one of the most important models within TAPCOM II.

---

[1]The discussion of direct-fire target selection in this chapter was prepared by Dr. Sam H. Parry and represents the results of his model development research efforts.

It should be noted that the decisions made apply to the elements within one aerial section as opposed to those made for an entire aerial maneuver unit. The latter decisions are formulated by the model reported in Chapter 4 and provide a frame of reference for the section decisions to be discussed in this chapter. The model reported in this chapter is designed to formulate the section decisions on a basis that is consistent with implementing any precedent unit decision.

To implement the movement and fire controller, subroutine HELCON has been developed and is reported in flow chart form in Volume 2. As discussed in Chapter 1, this subroutine is called from the DYNCOM main program each time an aerial vehicle section leader becomes the current element. It appears in the processing sequence just after the communications and intelligence models (Chapter 5) and just prior to the movement model (Chapter 7). Thus, at the time section movement and firing decisions are formulated, the decision element has access to current information about the enemy. Also, movement during the event can be in response to any new movement decisions made.

The reader should note that the decisions made apply to an entire aerial section. By convention, only one element from the section is processed by the decision model, and this element is taken as the section leader. Other elements in the section merely move and fire in response to the decisions that are made by the leader. To permit this scheme to work effectively, the section leader is always first to be processed among the elements of the section. This means that the clock times of the leader and the followers must be controlled so that the leader always becomes the current element before any other element in the section, and all other elements in the section must become current before the leader is again processed. This means also that any data describing a movement or firing decision that applies to an individual in the section must be recorded when the leader is the current element.

We will commence our discussion of the model by presenting definitions of some of the more important variables used by the model. We will then conclude the introduction by presenting an overview of the model structure. Finally, we will complete the chapter by presenting a description of each major processing step discussed in the model overview.

Basic Variables

The element that is processed by the model is ICE and is the leader of section ISEC. That is,

$$ICE = ISORG(1, ISEC).^1$$

Section ISEC is a member of maneuver unit MANUN and ICE is the leader of MANUN if ICE = MANLDR(MANUN). The variables ICE, ISEC and MANUN are taken from common area /ICECOM/.

Section ISEC has a corresponding aerial section number, NSEC, found by the relation NSEC = NAVSEC(ISEC). Similarly, the aerial unit number, NAT, is found from MANUN by the relation NAT= MANHEL(MANUN). Finally, aerial unit NAT is actually included in the fire-support structure of DYNTACS X as fire-support firer NF. The variable NF is found by the relation NF = NUMART + ITOTLN + NAT where NUMART and ITOTLN are the numbers of artillery and MISTIC firers, respectively (see common area /NUMBER/).

The variables that provide structure to the model consist of a set of movement state variables and movement decision variables defined for aerial unit NAT and aerial section NSEC. Most of these variables have been defined previously in either Chapter 1 or Chapter 4. We will review these variables below.

Unit Movement State Variables

The unit movement state variables define in detail the current movement activity being performed by aerial unit NAT. They were discussed in Chapter 1 and again in Chapter 4. The present movement activity of NAT is given by IUNACT(NAT) as defined in Table 6.1.

Examination of the entries in Table 6.1 reveals that NAT may be performing in one of four general ways; that is, NAT may

1. have no mission;

2. be performing a mission in which the objective is to engage targets with direct fire;

---

[1]In this chapter, unless otherwise noted, each array is contained in a common area bearing the array name. The common areas referenced appear in Volume 2.

Table 6.1

Unit Activities

IUNACT(NAT)	Activity
0	NAT is over the battlefield but is without a mission
1	NAT is performing a mission for a forward observer against a target of opportunity
2	NAT is performing a counterbattery attack mission for the artillery intelligence center
3	NAT is performing a search-and-destroy mission for the battlefield commander
4	NAT is performing a self-defense mission
5	NAT is performing an indirect-fire MISTIC launcher mission for the battlefield commander
6	NAT is performing a MISTIC forward observer mission for the battlefield commander
7	NAT is performing a counterbattery observation mission for the artillery intelligence center
8	NAT is performing a special forward observer mission for the battlefield commander
9	NAT is over the battlefield without a mission and is operating so as to protect itself from enemy fire
10	NAT is enroute to or enroute off the battlefield or is waiting for a mission while off the battlefield

3. be performing a mission in which the objective is to observe targets for some other element of the fire-support force; or

4. be performing a mission in which the objective is to engage targets with indirect fire.

Thus, we may introduce the variable MCLASS(NAT) to account for this more general breakdown. Table 6.2 shows the correspondence between MCLASS(NAT) and IUNACT(NAT).

Table 6.2

Correspondence Between MCLASS and IUNACT

MCLASS(NAT)	Corresponds to IUNACT(NAT) Values			
1	0	9	10	
2	1	2	3	4
3	6	7	8	
4	5			

The reader will recall from Chapter 1 that each movement activity for NAT commences with an enroute phase and is followed by a mission operations area phase. In each case, the enroute phase activity consists of the unit flying from the point at which the decision to commence the activity was made to an area in the vicinity of the point defined as the mission objective. During enroute movement, sections flying with the unit fly as part of an overall formation specified for the unit.

When the unit reaches the vicinity of the objective, the mission operations area phase commences. This area is defined as a circle with specified radius centered at the mission objective. The radius is defined from input data as will be discussed later in this chapter.

Activities of the sections that are operating with the unit during the mission operations area phase vary depending upon the type of activity being performed by the unit. Table 6.3 gives an operational description of the ground rules assumed by TAPCOM II. These activities will be discussed in much more detail in subsequent sections of this chapter. The important thing to note here is that the mission phase indicator coupled with the mission activity indicator, can be used to indicate what activities of sections operating with the unit are allowed. The mission phase indicator is defined as follows:

$$\text{IPHASE(NAT)} = \begin{cases} 1 & \text{if NAT is enroute to the mission operations area, and} \\ 0 & \text{if NAT is in the mission operations area.} \end{cases}$$

Unit Movement Decision Variables

As discussed in Chapter 4, movement decisions are formulated by the aerial unit decision element in the mission control model (subroutine AIRFB) and are implemented on a section basis in the section movement controller (subroutine HELCON). To provide a means of coordinating this process, two unit and one section movement decision variables are used. The variable MANACT(NAT) is set in subroutine AIRFB to indicate that a movement decision has been formulated for NAT. It takes on values that are greater by one than the mission activity indicator IUNACT(NAT) will be when the sections of the unit begin to perform the new mission; i.e., IUNACT(NAT) is set to MANACT(NAT) - 1 when the decision is implemented.

The maneuver unit leader is always the first element of the unit to react to the decision as will be discussed subsequently. Thus, when ICE is the maneuver unit leader, and MANACT(NAT) > 0, not only is it known that a decision has been made, but the nature of the decision is also known. The activity indicator is set as indicated above and then MANACT(NAT) is reset to zero to indicate that the unit has begun to react to the decision.

However, the decision has not necessarily been implemented by the entire unit. Though the section ISEC of which ICE is the leader has reacted to the decision, there may be other sections whose leaders have not been processed and consequently have not reacted to the decision. Thus, we use the variables LMUFL(MANUN) and ISACT(ISEC) to keep track of which sections have reacted to the decision and when the entire unit has reacted.

When the maneuver unit leader senses the decision, LMUFL(MANUN) is set to one. Also, ISACT(ISEC) is set to one for each section ISEC in MANUN. Then, as each section leader is processed and reacts to the decision,

Table 6.3

Section Activities During Objective Area Operations

IUNACT(NAT)	Section Activity
0	The entire unit occupies a loiter station and all sections fly in the unit formation
1	Each section operates independently, executing a search for targets and then engaging selected targets with direct fire
2	Each section operates independently, executing a search for the assigned enemy artillery battery and then engaging it when it is acquired
3	Each section operates as in the case of IUNACT(NAT) = 1
4	Each section operates as in the case of IUNACT(NAT) = 1
5	The entire unit occupies a loiter station and all sections fly in the unit formation except when delivering indirect MISTIC fire. A section with an indirect-fire assignment operates independently.
6	Each section operates independently executing a search for targets and then calling for indirect-fire as required
7	The entire unit executes a search for the assigned enemy artillery battery with each section flying in the unit formation
8	Each section operates as in the case of IUNACT(NAT) = 6
9	The unit operates as in the case of IUNACT(NAT) = 0
10	The unit operates as in the case of IUNACT(NAT) = 0

the corresponding entry in the array ISACT is reset to zero. When all entries have been reset, LMUFL(MANUN) is reset and the entire unit is then known to have reacted to the decision.

Section Movement State Variables

From Table 6.3, it is obvious that sections within the maneuver unit may operate independently or in the unit formation. Moreover, there are a variety of ways in which the sections can move independently. Therefore, we use several movement state variables to define in detail the exact movement activity being performed by a section. The first of these variables is JUNACT(NSEC) defined as in Table 6.4.

The reader will note from Table 6.4 that a section is considered to be operating with or in close proximity to other sections of the unit when JUNACT(NSEC) $\leq$ 3. When JUNACT(NSEC) $\geq$ 4, the section NSEC is operating in an entirely independent fashion. The separation between NSEC and other sections of the unit can be quite large.

The movement activity phase indicator for NSEC is JPHASE(NSEC) defined in general by the relationships

$$\text{JPHASE(NSEC)} = \begin{cases} 1 & \text{if NSEC is enroute, and} \\ 0 & \text{if NSEC has reached its destination.} \end{cases}$$

The exact connotation of a particular value for JPHASE(NSEC) is dependent both upon the value of JUNACT(NSEC) and upon the values of IUNACT(NAT) and IPHASE(NAT). A summary of the connotations used is presented in Table 6.5 while a more extensive discussion appears in the section describing route selection.

The final movement state variable for an aerial section is FORMSE(ISEC). This variable has been used since the initial version of DYNCOM to indicate the formation pattern number to be used by elements within a section. It is still used this way by aerial sections that are operating within, or in close proximity to, a larger unit formation. However, the convention is used in TAPCOM II that FORMSE(ISEC) is zero if ISEC is operating independently. When this is the case, another variable is used to indicate the section formation pattern and FORMSE(ISEC) is used merely to indicate whether or not ISEC is joining or is within the unit formation. Thus, if FORMSE(ISEC) > 0, section ISEC is either within or is joining a unit formation and the situation is possible if:

Table 6.4

Section Movement Activities

JUNACT(NSEC)	Section Activities
0	Aerial section NSEC is flying as part of the unit formation while the unit is enroute to its objective
1	Elements of aerial section NSEC are operating with the unit while the unit is in the mission operations area
2	Elements of aerial section NSEC are engaging targets with direct or indirect fire, or are illuminating targets for MISTIC attack
3	Elements of aerial section NSEC are waiting for indirect-fire support while operating as forward observers
4	Aerial section NSEC is retiring from the battlefield independent of the remainder of the unit
5	Aerial section NSEC is operating independent of the remainder of the unit and is flying so as to protect itself from enemy fire
6	Aerial section NSEC is rejoining the unit after having operated defensively independent of the remainder of the unit

Table 6.5

Section Movement Phase Indicator

JUNACT(NSEC)	JPHASE(NSEC)	Connotation
0	0	NSEC is considered a part of the unit enroute formation (applies only when IPHASE(NAT) = 1)
	1	NSEC is in close proximity to the unit enroute formation and is transitioning into the formation (applies only when IPHASE(NAT) = 1)
1	0	NSEC is either a part of the unit search or loiter formation or is searching for targets independently (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))
	1	NSEC is in close proximity to the unit search or loiter formation and is transitioning into the formation (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))
2	0	NSEC is engaging a target complex with direct or indirect fire or is illuminating targets for MISTIC attack (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))
	1	NSEC is moving toward the initial point in a route to be used while engaging a target complex with direct or indirect fire or while illuminating targets for MISTIC attack (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))

89

Table 6.5 (Continued)

JUNACT(NSEC)	JPHASE(NSEC)	Connotation
3	0	NSEC is occupying a loiter position while waiting for fire support to be delivered (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))
	1	NSEC is traveling toward a loiter position to be used while waiting for fire support to be delivered (applies only when IPHASE(NAT) = 0 and depends on IUNACT(NAT))
4	0	NSEC has retired from the battlefield and is no longer participating in the conflict
	1	NSEC is enroute off the battlefield and is operating independent of the unit
5	0	NSEC is occupying a loiter position while operating defensively, independent of the unit
	1	NSEC is enroute to an independent defensive position
6	0	Not allowed
	1	NSEC is enroute to join the unit after leaving an independent defensive position. NSEC has not reached the mission operations area

1. IPHASE(NAT) = 1 and JUNACT(NSEC) < 2, or

2. IPHASE(NAT) = 0, JUNACT(NSEC) < 2, and IUNACT(NSEC) = 0, 5, 7, 9 or 10.

If FORMSE(ISE) = 0, section ISEC is operating independently and is not a part of a unit formation.

Section Movement Decision Variables

There is only one section movement decision variable. The variable name is IRS and is used to indicate what type of route is to be selected by the section leader. The variable is initialized to zero at the very beginning of processing in subroutine HELCON to indicate that no route has been decided upon. Thereafter, it is passed among the various decision subroutines to be discussed in later sections, and is altered as required by the decisions made. It is used at the end of processing to indicate to the route selection model what type of route is to be selected. Then, it is even passed to the helicopter movement model (Chapter 7) to indicate what route, if any, has been chosen during the event. Thus, IRS is a very important variable of the movement and fire control model. The values of IRS permitted and the types of routes chosen in response to a given value of IRS are displayed in Table 6.6. Definitions of the various routes outlined in Table 6.6 are presented in much more detail in the route selection model description of a subsequent section.

Model Overview

Now that we have an operational definition of each of the basic variables used in the movement and fire controller, we can conclude the introduction with a general discussion of the major processing steps used within the model. Subsequent sections within the chapter will treat each step in much more detail.

The basic processing sequence within subroutine HELCON is as follows:

1. Perform an analysis to arrive at tentative movement and fire control decisions that are based upon movement alone.

2. If decisions were made in step 1, perform processing required to establish new values for the movement state and decision variables of the section and the unit. Establish new desired speeds and formation patterns as required.

Table 6.6

Section Movement Decision Variable

IRS	Route to Be Chosen
0	No route
1	Route to be used by a section transitioning into a unit formation
2	Route to be used by a section searching for targets
3	Route to be used by a section occupying a loiter station
4	Route arrays defining a loiter station are to be loaded
5	Route to be used by a section during the first phase of an attack
6	Route to be used by a section flying cross country
7	Route to be used by a section going to the initial point of an attack route
8	Route to be used by a section during the second phase of an attack
9	Route to be used by a section that is already flying in the unit formation
10	Route to be used by a section going to the initial point of a direct-fire attack route when the attack is not the first attack conducted against the specified target complex

3. If a maneuver unit leader, determine whether or not a new mission decision has been formulated for the unit. If so, perform processing required to initiate the implementation of the decision and to set the unit movement decision flag.

4. If the unit is performing a forward observer or MISTIC launcher mission, determine whether or not elements of the section can now commence active operations as forward observers or launchers. If so, perform processing required to activate elements of the section as forward observers or launchers.

5. Perform processing to arrive at new movement and fire control decisions for the section that are based upon tactical considerations and that supersede the decisions made in step 1.

6. If decisions were made in step 5, perform processing required to establish new values for the movement state and decision variables of the section and the unit. Establish new objectives, formations, speeds and axes of advance as required.

7. If the decisions made in step 5 were in response to a pending unit movement decision, determine whether or not all sections of the unit have now reacted to the unit decision. If so, reset the unit movement decision flag.

8. Select and record new routes for the section and the unit as required.

9. If elements of the section have been operating as forward observers or MISTIC launchers, determine whether or not they can continue such activity as a result of decisions made in steps 1 and 5. If not, perform processing to deactivate elements of the section as forward observers or launchers.

10. The processing of subroutine HELCON is complete.

The processing described above is accomplished by a complex of forty-five (45) subroutines including subroutine HELCON. Of these, eight have been described in detail elsewhere and will only be referred to briefly in our

discussion. The remaining thirty-seven (37) will be discussed in enough detail to familiarize the reader with their functions and to define the data sets upon which they operate.

Preliminary Decisions Due to Movement

As indicated by steps 1 and 5 in the processing sequence of the preceding section, we have arbitrarily classified all stimuli for movement and fire control decisions into two categories within TAPCOM II. This classification scheme is used merely for convenience. One set consists of stimuli produced as a result of movement conducted by the section. The other set consists of all other stimuli produced by changes in the tactical situation as viewed by the decision maker. The movement and fire control model formulates tentative decisions based upon stimuli from the first set, and then revises the decisions as required while analyzing stimuli from the second set. This model structure is arbitrary and is used only to reduce the complexity of the model. In this section, we discuss the tentative decisions made on the basis of section movement stimuli.

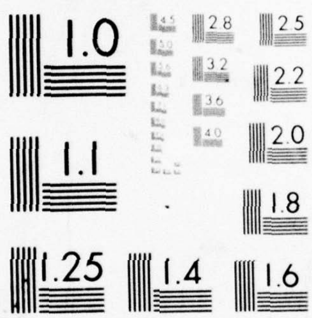
The movement that provides the stimuli upon which the decisions to be discussed are based is actually movement that occurred during the last events for the elements of the section. At the time that the section leader, ICE, becomes the current element, all elements of the section will have just completed an event. The previous event will have consisted of movement and firing activities that were based upon decisions formulated at the beginning of that event. Now, we are ready to formulate decisions that will guide our movement and firing activities during the current event. These decisions must account for movement that occurred in the last event.

The decisions of interest in this section are fomulated in subroutine FLGSET (see Volume 2). This subroutine not only formulates the decisions but also conducts special processing necessary to assure that the section implements the decision correctly. Thus, subroutine FLGSET performs steps 1 and 2 of the processing sequence of subroutine HELCON discussed earlier.

In general, when a movement decision is formulated, the only processing required to indicate that the decision has been made is to record a value for IRS to indicate that a new route for the section is required. However, because the section will be commencing a new route as a result of the decision, changes also occur in the section movement activity indicators JUNACT(NSEC) and JPHASE(NSEC). Moreover, it may also be necessary to change the unit movement activity indicators IUNACT(NAT) and IPHASE(NAT). Finally, it may be necessary to specify new formation patterns and to establish new values for the desired speeds of the section and the unit. We will discuss all these possibilities in the subsections below.

AD-A040 052 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL, (DYNCOM). VOLUME 1. HELICO--ETC(U)
DEC 71 G M CLARK DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-1(U) NL

3 OF 4
AD
A040052



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Decision Analysis

The decision analysis performed by subroutine FLGSET is summarized in Table 6.7. We have indicated the possible movement states of the section at the beginning of the event in the left-most columns. In the right-hand columns, we have indicated the value of IRS that is selected and the new values for the movement state variables.

The state variable IDPC is the desired position code for the current element and is contained in common area /ICECOM/. For helicopters, it is defined as follows:

$$\text{IDPC} = \begin{cases} 1 & \text{if the current element reached the end of the route recorded for the section during the last event, and} \\ 0 & \text{if otherwise.} \end{cases}$$

This variable is normally set only in the movement model (Chapter 7) and is always reset in subroutine FLGSET.

As may be seen in Table 6.7, we may nearly always use IDPC as the stimulus for a movement decision. This is because of the convention used in selecting routes within TAPCOM II. As will be discussed in a subsequent route selection discussion, routes of a specified length are recorded each time a section leader chooses a route. Then the movement model moves the elements of the section along the route until a decision to choose a new route is made. When the section reaches the end of its route, a new route must be selected. Thus, a movement decision is required and these are the types of decisions being discussed.

The only exceptions to this convention occur when a section operating in proximity to a unit is attempting to join the unit formation. Each time a route for this purpose is chosen, the end point of the route is the point in space that the section leader should occupy to be in the correct formation position. However, this point is always moving since the unit formation is moving. Thus, transition routes are chosen each event so long as the section continues the transition movement.

The material presented in Table 6.7 is easily understood if the reader will refer to the operational definitions of the movement state variables presented in Tables 6.1, 6.3, 6.4 and 6.5. The logic of the decision analysis follows directly from those definitions and the definitions of IRS presented in Table 6.6. There are, however, three procedures that must be clarified. One is that which is used when the section arrives at a defensive position while

Table 6.7

Section Decision Analysis

Beginning State					Final State				Route
JUN	JPH	IDPC	IUN	IPH	JUN	JPH	IDPC	IPH	IRS
0	0	1	0,5,9,10	0,1	1		0	0	3
0	0	1	1-4,6-8	0,1	1		0	0	2
0	1	0	0-10	0,1					1
0	1	1	0-10	0,1		0	0		9
1	0	1	0,5,9,10	0,1			0		4
1	0	1	1-4,6-8	0,1			0		2
1	1	0	0-10	0,1					1
1	1	1	0-10	0,1		0	0		9
2	0	1	0-10	0,1			0		8
2	1	1	0-10	0,1		0	0		5
3	0	1	0-10	0,1			0		4
3	1	1	0-10	0,1		0	0		3
4	0	1	0-10	0,1			0		4
4	1	1	0-10	0,1		0	0		0
5	0	1	0-10	0,1			0		4
5	1	1	0,9	0	1		0		1
5	1	1	0,9 / 1-8,10	1 / 0,1		0	0		3
6	1	1	0,5,7,9,10	0,1	1		0		1
6	1	1	1-4,6,8	0,1	1	0	0		2

Legend

JUN = JUNACT(NSEC)

JPH = JPHASE(NSEC)

IUN = IUNACT(NAT)

IPH = IPHASE(NAT)

Blanks indicate no change in state variables

operating independently (JUNACT(NSEC) = 5, JPHASE(NSEC) = 1). The reader will recall that a defensive position is a point over the battlefield at which the section loiters. This position is intended to afford protection from enemy fire. It is possible that the remainder of the unit already occupies the same position. This is true if NAT is operating defensively (IUNACT(NAT) = 9) or is waiting for a mission (IUNACT(NAT) = 0) and has achieved its area of operations (IPHASE(NAT) = 0). As explained in Chapter 4 , all sections operating defensively occupy the same position, and if the unit itself operates defensively or begins to wait for a mission, the same position will also be selected. Thus, when the case arises, the arriving section merely joins the unit formation instead of loitering independently.

Another procedure that must be clarified is the one in which a section is retiring independently and reaches the end of the route (JUNACT(NSEC) = 4, JPHASE(NSEC) = 1). No route is chosen for the section in this case since the indication is that the section has retired. The movement model will remove all elements of the section from the battle when this is true so no processing for a new route is required.

Finally, the reader will note that the model is designed to exclude the case in which JUNACT(NSEC) = 6 and JPHASE(NSEC) = 0. Such a case has no meaning and is impossible. The reader will also note that some of the other initial combinations of JUNACT(NSEC) and IUNACT(NAT) are also impossible but are included in the table for completeness. For example, it is possible to have JUNACT(NSEC) = 2 only for IUNACT(NAT) = 1-6 even though all possible values for IUNACT(NAT) are shown in the table.

Formations and Speeds

As will be discussed in more detail in a subsequent section of this chapter, TAPCOM II is designed to allow dynamic selection of speeds and formations to be used by sections that are operating independently. The formations and speeds used by sections within a maneuver unit formation are also selected dynamically. In all cases, the section leader is the element that performs the selection process. Moreover, the formations and speeds selected are dependent upon whether or not the section is operating independently. Therefore, special processing is required to set the speed and formation of a section that formulates movement decisions.

Of the situations discussed previously, only two necessitate selection of formations and speeds. These situations occur when:

a. a section begins to loiter independently or with the unit,

b. a section begins to search for targets independently or with the unit.

In all other cases, the speed and formation being employed continue to be used since the variables already have their proper values for the new movement activity. This can be seen by analysis of Table 6.7.

The two subroutines used to set the formation and speed in the above situations are HFORM and SECPRM. The former is used when a section is joining a unit formation while the latter is used when the section is beginning an independent operation. Both subroutines will be discussed in more detail in the subsequent discussion.

Both subroutines have a calling parameter IFTN that specifies the type of operation being performed. For our purposes here, the definition of IFTN is as follows:

$$IFTN = \begin{cases} 2 & \text{if processing is for a loiter operation, and} \\ 3 & \text{if processing is for a target search operation.} \end{cases}$$

In Table 6.8, we have summarized the cases which require the setting of formation and speeds. In the left-most columns, we have specified the beginning state variables in a manner similar to that used in Table 6.7. In the right-hand columns, we have indicated the subroutine used and the value of IFTN. The reader will again note that certain combinations of JUNACT(NSEC), IUNACT(NSEC), and IPHASE(NSEC) shown in the table are impossible. They are included only for completeness. Note also that the processing indicated for sections arriving at defensive positions (JUNACT(NSEC) = 5) is in agreement with the procedures outlined earlier. That is, if the unit is already at the position (IUNACT(NAT) = 0 or 9 and IPHASE(NAT) = 0), the section is processed to join the unit instead of loitering independently. Note that two values of IUNACT(NAT) are possible in this situation since the unit may be operating defensively or waiting for a mission at the position as explained previously.

Selecting a Maneuver Unit Leader

As explained in Chapter 1, a convention used in TAPCOM II is to require that the element designated as the leader of an aerial maneuver unit be operating with the unit. That is, the leader element LDR defined by the relation LDR = MANLDR(MANUN) cannot be the leader of a section that is operating independently (going or coming from a defensive position or retiring). Thus, LDR must be a member of an aerial section NSEC with JUNACT(NSEC) less than or equal to three.

Table 6.8

Selecting a Formation and Speed

Beginning State					Subroutine Used		
JUN	JPH	IDPC	IUN	IPH	IFTN	HFORM	SECPRM
0	0	1	0,5,9,10	0,1	2	x	
0	0	1	7	0,1	3	x	
0	0	1	1-4,6,8	0,1	3		x
3	1	1	0-10	0,1	2		x
5	1	1	0,9	0	2	x	
5	1	1	0,9	1	2		x
			1-8,10	0,1			
6	1	1	0,5,7,9,10	0,1	2	x	
6	1	1	1-4,6,8	0,1	3		x

Legend:

JUN = JUNACT(NSEC)
JPH = JPHASE(NSEC)
IUN = IUNACT(NAT)
IPH = IPHASE(NAT)

In a later discussion in this chapter, we outline the processing that is required to choose a new leader when the present leader leaves the unit. Here, we must deal with the situation that occurs when a section rejoins the unit. We must determine if the leader of the returning section should be the maneuver unit leader.

First, the cases in which a section rejoins the unit are:

1. when a section reaches the mission operations area after returning from an independent defensive position (beginning state variables: JUNACT(NSEC) = 6; JPHASE(NSEC) = 1; IDPC = 1); and

2. when a section arrives at an independent defensive position and the unit already occupies the position (beginning state variables: JUNACT(NSEC) = 5; JPHASE(NSEC) = 1; IDPC = 1; IUNACT(NAT) = 0 or 9; IPHASE(NAT) = 0).

When one of these cases occur, subroutine NATLDR is called to designate the proper maneuver unit lead element LDR. If LDR = ICE, then ICE is designated as the new maneuver unit leader; i.e., MANLDR(MANUN) = ICE.

Subroutine NATLDR uses the basic TAPCOM II assumption that the leadership hierarchy within an aerial maneuver unit is as follows:

1	leader of section one in platoon one,
2	leader of section two in platoon one,
3	leader of section one in platoon two,
4	leader of section two in platoon two,
⋮	
2N-1	leader of section one in platoon N,
2N	leader of section two in platoon N.

Of course, it is possible that the maneuver unit organization has been specified initially with missing sections and platoons. However, it is also possible that as many as fourteen elements would exist in the hierarchy because there are as many as two sections in a platoon and as many as seven platoons in the unit ($N \leq 7$).

The processing scheme consists of successively analyzing the elements in the leadership hierarchy and designating the leader as the first element found that satisfies the leadership criterion. That is, LDR is the first element found for which JUNACT(NSEC) < 4 where NSEC is the aerial section number of the element. Details of the scheme can be seen in the flow chart of NATLDR

in Volume 2. The reader should note the schemes used for specifying formation positions within TAPCOM II are designed to account for missing sections within a unit as will be explained. Thus, the process of designating a new unit leader does not affect the process of specifying formation positions of elements within the unit. The new leader assumes the lead position of the formation and other element positions are adjusted automatically to account for the change in leadership.

Unit Arrival in Operations Area

Special processing must be performed when a unit arrives in its mission operations area after enroute movement. This situation is indicated when a section has the beginning state variables:

JUNACT(NSEC) = 0
JPHASE(NSEC) = 0, and
IDPC = 1.

The first two values indicate that section NSEC has been operating in the unit enroute formation (Table 6.5). The last value indicates that the end of the section route (and the unit route) has been reached.

First of all the time at which the unit arrived must be recorded. This variable is TMISUN(NAT) and its value is established by the relation:

TMISUN(NAT) = CLOCKT

where CLOCKT is the clock time of the current element ICE as recorded in common area ICECOM. The variable TMISUN(NAT) is used in determining when the unit can terminate its mission as explained in Chapter 4. It is also used in a counterbattery observation mission procedure that was referred to in Chapter 4 and will be explained in more detail in a subsequent discussion in this chapter.

The other processing required when a unit arrives in a mission operations area is the performance of a procedure to insure that all sections flying with the unit react properly and in a timely fashion to the unit transition. The procedure consists of setting the desired position codes, LDPC(JK), of the leaders, JK, of all the sections operating with the unit so that they will react to the unit transition during their next event. Then their movement state variables are set to values that will result in their reacting properly. Note that when a leader, JK, becomes the current element, then LDPC(JK) becomes IDPC as defined previously.

The sections for which the procedure outlined above applies are those sections that have been flying with, or transitioning into, the unit enroute formation. From previous definitions, these sections are those for which FORMSE(ISEC) > 0. Another group exists in the special case in which the unit has reached a defensive position or a loiter station at which it is to await a new mission (IUNACT(NAT) = 0 or 9). Then, those sections that are members of the unit and are in independent defensive positions (JUNACT(NSEC) = 5 and JPHASE(NSEC) = 0) are included. This procedure is required because of the TAPCOM II convention of specifying the same position for defensive and waiting loiter stations both for the unit and for all sections within the unit. Therefore, sections operating defensively should be made to realize that the unit has reached the same position so that they can join the unit formation.

The movement state variables of the above sections are set as follows: JUNACT(NSEC) = 6, JPHASE(NSEC) = 1. The reader may verify from Table 6.7 that these values will insure that the sections react to the unit transition in the proper way. The desired position code for a noncurrent element LD is LDPC(LD). When LD becomes the current element the value for LDPC(LD) will be loaded into IDPC in common area /ICECOM/.

New Mission Processing

Step 3 in the processing sequence of subroutine HELCON is performed if the current element is the maneuver unit leader (ICE = MANLDR(MANUN)). The processing is concerned with recording the fact that a new unit mission decision has been formulated in the mission control model of Chapter 4 and with establishing controls that assure that each section in the unit reacts to the decision. The latter processing is required since the decision is formulated in the mission control model, and the decision must be implemented on a section basis in the movement and fire control model. The processing is accomplished in subroutine NEWMIS whose detailed processing sequence is illustrated by the flow chart in Volume 2.

As explained in the introduction to this chapter, the fact that a new mission decision is pending is indicated by MANACT(NAT) > 0. The value of MANACT(NAT) is always greater by one than the value of IUNACT(NAT) will be for the new mission. For example, if MANACT(NAT) = 4, the new mission is a search-and-destroy mission for the battlefield commander (IUNACT(NAT) = 3). Thus, the first processing in subroutine NEWMIS is the recording of the new value for IUNACT(NAT) and the resetting of MANACT(NAT); i.e., IUNACT(NAT) = MANACT(NAT)-1 and then MANACT(NAT) = 0. If MANACT(NAT) = 0 upon entry, no processing is required.

After the new mission indicator is set, the unit and section decision flags are set so that the sections will react to the unit decision. The processing is accomplished in subroutine LMUSET. As explained in the introduction, LMUFL(MANUN) is set to one to indicate that a decision is being implemented by sections of the unit. Then the decision flags, ISACT(ISEC), are set to one for each section, ISEC, in the unit. The only sections that are excluded are those that have retired independently or those that are casualties. These sections are inactive and cannot react to the decision.

Actually, subroutine LMUSET is also used to reset the unit decision flag LMUFL(MANUN) when all sections have reacted to the decision. In this case, all sections ISEC in the unit are examined to determine the value that exists for ISACT(ISEC). If all values of ISACT(ISEC) are zero, then all sections have reacted and LMUFL(MANUN) can be reset. References to this use of subroutine LMUSET will be made in subsequent discussions. The input parameter ICNT controls the mode of operation of subroutine LMUSET; i.e.,

$$\text{ICNT} = \begin{cases} 1 & \text{if the section and unit flags are to be set, and} \\ 0 & \text{if the unit flag is to be reset.} \end{cases}$$

The next processing in subroutine NEWMIS sets the mission class indicator MCLASS(NAT) in a fashion indicated by the definitions appearing in Table 6.2. The value is dependent upon IUNACT(NAT). Next, IPHASE(NAT) is set to one to indicate that the unit is commencing enroute movement.

Finally, subroutine SECSET is used to establish all the movement and organization variables to be used by the unit and the sections flying with the unit formation during the enroute phase of the new mission. The variables that are set are discussed in the paragraphs which follow.

Axis of Advance

The number of the axis of advance is incremented by one to indicate that a new axis of advance is being used; i.e.,

$$\text{NAXIS(MANUN)} + 1 \rightarrow \text{NAXIS(MANUN)}.$$

Normally, the new value will be one since NAXIS(MANUN) is usually zeroed by the mission assignment procedure of Chapter 4. However, the new value can be as large as four when the new mission is actually movement between indirect-fire MISTIC launcher stations. See Chapter 4 for clarification.

Objective Position

The objective position is recorded by the coordinates OBJX(MANUN) and OBJY(MANUN). These values are taken from the variables XD(NF) and YD(NF), respectively. The latter values are specified by the model of Chapter 4 upon assignment of the mission and represent the coordinates of the center of the new mission operations area. The reader will recall that NF is the fire-support firer number of aerial unit NAT. The variables OBJX(MANUN) and OBJY(MANUN) are recorded simply to provide consistency of notation between TAPCOM II and the armor module of DYNCOM.

Formations and Speeds

Formation patterns and desired speeds for the unit, and for all sections and platoons within the unit, are recorded in subroutine HFORM. This subroutine has been referred to previously and will be discussed in detail in a subsequent section. The calling parameter used is IFTN and its value is one to indicate that data for enroute unit movement are to be recorded.

Direction

Unit movement direction is recorded as DIRMU(MANUN). This variable is used only once in TAPCOM II but is recorded for notational consistency with the armor module. The computation is

$$\mathrm{DIRMU(MANUN)} = \tan^{-1}\left(\frac{YT - YLD}{XT - XLD}\right)$$

where

XT, YT = OBJX(MANUN), OBJY(MANUN), and

XLD, YLD = coordinates of the maneuver unit leader (ELOCX(LDR), ELOCY(LDR)).

Processing for New Units

The final processing in subroutine SECSET is concerned with units that are receiving their initial mission assignment. Such units have been awaiting a mission while off the battlefield (IUNACT(NAT) = 10 and IPHASE(NAT) = 0). An indication that such a unit is being analyzed is LIMOV(LDR) = 0 where LDR is the maneuver unit leader. The variable LIMOV(I) is defined as follows:

$$\text{LIMOV(I)} = \begin{cases} 0 & \text{if element I has never moved from its initial position, and} \\ 1 & \text{if otherwise.} \end{cases}$$

When a unit is coming onto the battlefield for the first time, data must be initialized for each element in the unit to indicate the element's position, speed, direction and movement status. Such processing is required since elements in the unit have been inactive previously. The data items mentioned above are not initialized for helicopter elements in these inactive units. We will discuss each data item in the following paragraphs. The data applies to each element NELE that is a member of the oncoming unit.

Speed

The speed of NELE is recorded as ESPD(NELE) and is set from the desired speed of the maneuver unit; i.e.,

ESPD(NELE) = SPDMU(MANUN).

The desired maneuver unit speed is set in subroutine HFORM.

Direction

The direction of motion of NELE is recorded as EDIR(NELE) and is set from the maneuver unit direction of motion; i.e.,

EDIR(NELE) = DIRMU(MANUN).

The reader will recall that DIRMU(MANUN) was discussed in a preceding paragraph.

Movement Status

The movement status of NELE is indicated by the variables LIMOV(NELE) and LMOVF(NELE). The variable LIMOV(NELE) has been discussed previously and indicates whether or not NELE has ever moved. In this case, LIMOV(NELE) is set to one.

The variable LMOVF(NELE) is defined as follows:

$$\text{LMOVF(NELE)} = \begin{cases} 0 & \text{if element NELE did not move in its last event, and} \\ 1 & \text{if otherwise.} \end{cases}$$

Obviously, as represented in TAPCOM II, helicopters move during every event until they leave the battlefield. Consequently, LMOVF(NELE) is set to one.

Position

The position of any DYNCOM vehicular element NELE is given by the battlefield coordinates X, Y, and Z. The Z position is measured relative to an arbitrary zero elevation plane, and for ground vehicles, is simply the elevation of the terrain at X and Y. However, helicopters can be above the terrain. Therefore, an increment to account for this altitude must be added to the terrain elevation at X and Y to determine the Z position. The helicopter's altitude above the terrain is ELOCZ(LHCE), where LHCE is NELE's helicopter number; viz., LHCE = LHICE(NELE). The X and Y coordinates of any element are recorded as ELOCX(NELE) and ELOCY(NELE), respectively. (See Chapter 9 and Volume 2 for a description of how function ELVATE has been modified to yield the Z position of any vehicle.)

The initial position coordinates of NELE in X, Y and Z are computed by the relations:

$$X = XLD + DELX * \cos(BETA) - DELY * \sin(BETA),$$
$$Y = YLD + DELX * \sin(BETA) + DELY * \cos(BETA), \text{ and}$$
$$Z = ZLD + DELZ.$$

The coordinates XLD, YLD, and ZLD are the coordinates of the maneuver unit leader which are input to the simulation. That is, ELOCX(LDR), ELOCY(LDR), and ELOCZ(LHICE(LDR)) are input for the maneuver unit leader LDR. They should be selected by the simulation user as the position desired for the maneuver unit when it becomes active.

The angle BETA is computed by the relation:

$$BETA = DIRMU(MANUN) - \pi/2$$

while the increments DELX, DELY, and DELZ are computed by the relations:

$$DELX = DELX1 + DELX2,$$
$$DELY = DELY1 + DELY2, \text{ and}$$
$$DELZ = DELZ1 + DELZ2.$$

They represent the position in X, Y and Z, respectively, of NELE relative to the maneuver unit leader LDR.

The components DELX1, DELY1, and DELZ1 are the offsets of NELE's section leader NSAVE relative to LDR while DELX2, DELY2, and DELZ2 are the offsets of NELE relative to NSAVE.

All of the offset components defined above are computed in subroutine OFFSET. This subroutine is designed to compute formation positions from input formation information and the formation pattern number determined in subroutine HFORM or subroutine SECPRM. The computations also account for formation position adjustments required because sections and/or platoons have left the unit formation. We will discuss the details of the subroutine in a subsequent section.

Forward Observer and Launcher Processing

Steps 4 and 9 of the processing sequence of subroutine HELCON are best described together as they are both associated with special processing required to control the activities of helicopter sections belonging to a unit performing a forward observer mission or an indirect-fire MISTIC launcher mission for the battlefield commander. For the most part, this processing is associated with activating or deactivating the forward observer or launcher function of such sections. A general description of the overall model design will facilitate understanding of the details of processing that will be presented in subsequent paragraphs of this section.

The need for fire support in the form of aerial MISTIC launchers or forward observers is assessed in the target selection model described in reference 1. Requests for support of these types are generated by the battlefield command team element in subroutine AFSC and are transmitted over the ground-to-air radio net. The message is processed and the mission data become available at a time computed by subroutine AFDC (see reference 2). The message is addressed to a specified aerial unit, and if the request is for a MISTIC launcher or MISTIC forward observer mission, the MISTIC unit to which the aerial unit will be assigned is recorded. Subroutine AIRFB (Chapter 4) then performs processing in which the mission is actually assigned to the specified aerial unit. The number of the MISTIC unit to which a unit performing a MISTIC launcher or forward observer mission is assigned is made available through the variable KFO(NF) where NF is the fire-support firer number of the aerial unit as defined previously.

Elements of the unit do not become active as launchers or forward observers until the unit reaches its mission operations area. Then, if a forward observer mission is being conducted, each section that is flying with the unit is designated as a forward observer team. A forward observer element is

created and corresponds to the aerial section. The new element becomes active and commences operations. Subroutine AFO described in reference 1 processes the element as it would any other forward observer. Operations of the forward observer element continue until the aerial section decides to retire or seek a defensive position or until the unit terminates the mission.

The same type of processing is used if a MISTIC launcher mission is being flown. In this case, each aerial section is designated as a MISTIC launcher and an indirect-fire launcher fire-support element is created. This element becomes active and commences operations with required processing being conducted in subroutine MFB as described in Chapter 3. The vehicle within the section that is to actually perform launch operations is designated when the fire-support element is created. Thereafter, the MISTIC ammunition supplies of elements within the section are continually monitored and the element to perform the launch operations is redesignated as needed. The element with the most MISTIC ammunition remaining is always designated as the launcher.

Subroutine NEWFO performs the processing designated as step 4 of the HELCON processing sequence. This subroutine creates the fire-support elements described above and designates the element to perform launch operations if needed. Subroutine DARFO performs the analysis designated as step 9 of the processing sequence. Fire-support elements are deactivated as required within this subroutine. We will discuss the details of these subroutines in the following paragraphs.

Creating a Fire-Support Element

According to the rules discussed above, the launcher and forward observer elements are not activated until a unit reaches the operations area. Then each section that is operating with the unit is designated as a fire-support element. Thus, the criteria for designating a section as a forward observer or launcher are as follows: IUNACT(NAT) = 5, 6 or 8 and IPHASE(NAT) = 0 and JUNACT(NSEC) < 4.

When one of the above relations holds, then a fire-support element must be created and activated, and a correspondence between the aerial section and the new element must be established. The variables that must be initialized are summarized below.

NUM	The launcher or forward observer number associated with the aerial section (COMMON/ICECOM/).
KFUNC	The code that specifies whether the section is performing as a forward observer or launcher (1 - launcher, 2 - forward observer)(COMMON/ICECOM/).

$$KFUNC = \begin{cases} 1 & \text{if IUNACT(NAT) = 5, and} \\ 2 & \text{if IUNACT(NAT) = 6 or 8.} \end{cases}$$

IMIST(NUM) — The MISTIC unit to which an aerial section is assigned as MISTIC forward observer team NUM.

IMIST(NUM) = KFO(NF) - ITOTFO - NUMART where NF = NUMART + ITOTLN + NAT.

NMISUN(NUM) — The MISTIC unit to which an aerial section is assigned as launcher number NUM.

NMISUN(NUM) = KFO(NF) - ITOTFO - NUMART where NF = NUMART + ITOTLN + NAT.

NOBVH(KSUB) — The vehicle designated as the carrier for FO team NUM (KSUB = NUM) or the vehicle that is to perform the launch operations (KSUB = ITOTFO + NUM).

IFRFL(KSUB) — The fire-support suspension indicator (0 - fire-support activities permitted; 1 - fire-support activities suspended).

IFRFL(KSUB) = 0.

ECLOCK(NFCLK) — The clock of the fire-support element (NFCLK = NTFO + NUM for FO's; NFCLK = NTFB + NUMART + NUM for launchers).

ECLOCK(NFCLK) = CLOCKT (current clock time of current element ICE from COMMON/ICECOM/).

TIFRDY(NUM) — The time at which indirect-fire activities can be commenced by MISTIC launcher element NUM.

TIFRDY(NUM) = CLOCKT.

KFOD(NUM) — The fire-support activity code for FO element NUM.

KFOD(NUM) = 0.

IFBMIS(NF) The fire-support activity code for MISTIC launcher element NUM (NF = fire-support firer number of NUM).

IFBMIS(NF) = 0.

The variable NOBVH(KSUB) is initialized as ICE. Then, if the unit is performing a MISTIC launcher mission, the procedure discussed in a subsequent paragraph for launcher reassignment is used to redesignate the launcher vehicle if required. Otherwise, NOBVH(KSUB) remains as initialized.

The launcher or FO number NUM is more difficult to determine than the above variables. Moreover, the procedure used assumes that the simulation user has followed the scheme outlined below in preparing the initial data set.

From Chapter 1, there are a total of ITOTLN MISTIC launchers and a total of ITOTFO FO's. Of the FO's, the first NTSFO are special FO's as defined in Chapter 2. All three of the above constants are contained in common area /NUMBER/.

Now, the initial data set must be set up to allow aerial sections to be given FO and launcher numbers during the course of the battle. This means that launcher and FO numbers must be available which are not being used by any elements at the time that aerial sections begin acting as launchers or FO's. Therefore, the user should prepare the initial data set in a manner that is illustrated by the example in Table 6.9. In the data set, room is allowed for one MISTIC FO, two special FO's, and two MISTIC launchers. The other FO's and launchers are active at the start of the battle.

In practice, a very important question is how much room should be left for aerial FO's and launchers. In theory, all the aerial sections could be assigned the FO or launcher function during the battle. On the other hand, a battle could be run to conclusion without an aerial FO or launcher assignment. In the first case, simulation errors could arise if too little room has been allocated. In the second case, a significant set of unused data could result. During simulation check out, we have left room for as many FO's and launchers as there are aerial <u>units</u> (not aerial <u>sections</u>) being represented. This approach avoids excessive unused data, and simulation errors have not been encountered. The user should monitor his simulation runs carefully to assure that the numbers he is using are satisfactory.

The procedure based on the assumed input data scheme for selecting a launch number for an aerial section is as follows:

Table 6.9

Sample Data

Description	I	NUM	NOBVH(I)	IMIST(NUM)	INART(NUM)	NMISUN(NUM)	LFUNC(K)
Special Red FO Bravo	1	1	0	0	2	---	---
Reserved	2	2	0	0	0	---	---
Reserved	3	3	0	0	0	---	---
Special Blue FO	4	4	15	0	1	---	2
Special Red FO	5	5	47	0	2	---	2
Reserved	6	6	0	0	0	---	---
Regular Blue MISTIC FO	7	7	19	1	0	---	2
Regular Red MISTIC FO	8	8	23	2	0	---	2
Regular Blue Artillery FO	9	9	5	0	1	---	2
Regular Red Artillery FO	10	10	52	0	2	---	2
Blue Launcher	11	1	17	---	---	1	1
Reserved	12	2	0	---	---	0	---
Reserved	13	3	0	---	---	0	---
Red Launcher	14	4	34	---	---	2	1

* NTSFO = 5
ITOTFO = 10
ITOTLN = 4
NUMBLU = 20
NUMELE = 60
NBART = 1
NUMART = 2
NBMIS = 1
MISTUN = 2

INART(NUM) - artillery unit to which FO NUM is assigned
LFUNC(NUM) - corresponds in definition to KFUNC
K = NOBVH(I)

* See COMMON/NUMBER/ for definition of all constants in this column.

Analyze launcher numbers J = 1,...ITOTLN. Designate as NUM the first launcher number for which NOBVH(I) = 0 where I = ITOTFO + J.

The procedure for selecting the FO number is similar to the above procedure. However, FO numbers J=1,...,NTSFO are analyzed if the section is to perform as a special FO team while numbers J = NTSFO + 1,...,ITOTFO are analyzed if the section is to perform as a MISTIC FO team. NUM is selected as the first value of J for which NOBVH(J) = IMIST(J) = INART(J) = 0.

<u>Redesignating a Launcher</u>

As mentioned previously, the section leader is initially designated as the vehicle to conduct MISTIC launch operations when indirect MISTIC fire is requested. However, the situation is reevaluated after initialization and continues to be reevaluated on each subsequent event of the section leader. The element in the section with the greatest remaining supply of MISTIC ammunition is desired as the launcher.

The procedure used each event is very straightforward. Each element I in the section is examined and his MISTIC ammunition supply is determined. The element in the section with the largest supply is selected.

The ammunition code of the MISTIC round is ITYP = IFMC(LCOD) where LCOD is the MISTIC unit weapon code; i.e.,

LCOD = LWCOD(NUMELE+NUMART+NMIS) - MWART

where NMIS = NMISUN(NUM) and the other constants are as defined previously or in common area /NUMBER/. The supply of ammunition code ITYP is found by subroutine AMMO.

If all elements in the section have depleted their missile supplies, then IFRFL(NUM + ITOTFO) is set to one to indicate that launcher NUM is no longer active. Otherwise, the selected element JSAVE is designated as the vehicle to perform launch operations. The steps are:

LNUM(JSAVE) = NUM
LFUNC(JSAVE) = 1
NOBVH(ITOTFO + NUM) = JSAVE.

Note that NUM in COMMON/ICECOM/ for the leader is not zeroed. This procedure is used to permit more convenient processing in other parts of TAPCOM II. It does not really affect the reassignment process. Note also that the above processing is not conducted if the section is in the midst of a launch

operation at the time of analysis. This convention alleviates difficulties that might arise if redesignation occurred while a launcher element were actively operating in subroutine MFB. The section is conducting firing if JUNACT(NSEC) = 2.

Fire Fight Flags for FO's

One final bit of processing occurs in subroutine NEWFO when an FO team is created and thereafter during each event of a section with the FO designation. The fire fight flag for FO NUM is set. This flag is NSTHFF(NUM) and its use is discussed in reference 1.

By definition, a special FO team always considers an intense fire fight to exist. Therefore, if IUNACT(NAT) = 8, NSTHFF(NUM) is set to one and remains one for the duration of the mission.

For MISTIC FO teams, the fire fight flag is determined according to the procedure outlined in reference 1. We summarize the procedure here.

1. Compute a subscript K = 2(KOLOR + 1) where

$$\text{KOLOR} = \begin{cases} 0 & \text{if NUM is blue, and} \\ 1 & \text{if otherwise.} \end{cases}$$

2. Call subroutine ISTHFF to compute FFRAT, the ratio of known enemy firers to friendly survivors.

3. If FFRAT exceeds the fire fight threshold, set the intense fire fight flag; i.e.,

$$\text{NSTHFF(NUM)} = \begin{cases} 1 & \text{if FFRAT} > \text{CF(K), and} \\ 0 & \text{if FFRAT} \leq \text{CF(K).} \end{cases}$$

Deactivating Launchers and FO's

At the end of subroutine HELCON, after all decisions are made for the event (step 9), subroutine DARFO is called to determine whether or not an aerial section should be deactivated as an aerial FO team or launcher. Obviously, if the FO or launcher number NUM in common area /ICECOM/ is zero, no processing is required since the section is not operating as an FO team or launcher. Otherwise, processing must occur.

First of all, it is determined whether or not the section is still operating in the mission operations area. If so, the section can continue the FO or launcher activity. Thus, the conditions for continuing are: JUNACT(NSEC) = 2 or 3 or JUNACT(NSEC) = 1 and IUNACT(NAT) = 5, 6, or 8. The first condition specifies that the section is currently engaged with a target. The second condition specifies that the section's unit is still performing the mission and the section is operating with the unit.

If the FO or launcher should be deactivated, the following processing occurs:

Launchers--Set NMISUN(NUM) = 0
Set NOBVH(ITOTFO + NUM) = 0

FO Teams--Set IMIST(NUM) = 0
Set NOBVH(NUM) = 0

Both--Set ECLOCK(NFCLK) = ∞ where NFCLK is the clock number of NUM

Set LNUM(NELE) = 0 for each element in section NSEC

Set LFUNC(NELE) = 0 for each element in section NSEC

Set KFUNC = 0 and NUM = 0 (common area /ICECOM/ for the section leader).

Final Tactical Decisions

As discussed in a previous section, we have arbitrarily classified all stimuli for movement and fire control decisions into two categories within TAPCOM II. One set consists of stimuli produced as a result of movement conducted by the section. The other set consists of all other stimuli produced by changes in the tactical situation as viewed by the decision maker. The movement and fire control model formulates tentative decisions based upon stimuli from the first set, and then revises the decisions as required while analyzing stimuli from the second set. The latter procedure occurs at step 5 of the processing sequence for subroutine HELCON and will be discussed in this section. We will also discuss the processing that occurs as steps 6 and 7 of the processing sequence of subroutine HELCON since it is conducted to implement the decisions formulated at step 5.

Examples of the type of decisions that are based upon stimuli other than movement are listed below:

1. The section decides to retire independently from the battlefield.

2. The section decides to seek a defensive position independent of the unit.

3. The section decides to engage a target with direct fire.

4. The section decides to commence movement required to engage a target with indirect MISTIC fire.

5. The section decides to commence movement required to illuminate targets for indirect MISTIC fire delivered by some other element.

6. The section decides to commence movement to a waiting position to await delivery of requested fire support.

7. The section decides to commence movement that is consistent with a movement decision formulated by the maneuver unit leader.

Obviously, the decisions cited above are not all available to a section at a given moment. The tactical situation that exists determines the subset of decisions that may be considered. Then, within a given subset, a definite hierarchy exists among the available decisions. Within TAPCOM II, a careful logical analysis has been conducted to arrive at the contents of the various decision subsets and then to construct the hierarchy among the available decisions in each subset.

The major assumptions used in the decision analysis are outlined below:

1. A section that is presently conducting a firing run must continue the firing run. The section cannot react to movement decisions made by the maneuver unit leader until the run is completed. The rationale behind this assumption is that a section conducting firing activities would be too involved to react to new decisions until the firing is completed.

2. A section that is retiring from the battlefield independent of the unit must continue this movement activity and cannot react to unit movement decisions. This assumption is based upon the fact that a retirement decision is made only when the section can no longer function as an effective organization.

3. A section that is not retiring independently or conducting a firing run must consider the decision to retire above all other decisions. The decision is automatic when the unit itself has decided to retire. Justification of this assumption is provided by the rationale of assumption 2.

4. If the retirement decision is considered and rejected, the section will then consider any movement decision made by the maneuver unit leader above all other decisions. This assumption is made to insure that all sections within the unit react as quickly as possible to new decisions made by the maneuver unit leader. All sections that are not retiring independently eventually react to each unit decision.

5. Given that no maneuver unit decision is pending, the section will make movement and firing decisions that are consistent with the present movement state of the section. In general, the decisions available are:

   a. continue the present movement activity,

   b. select a direct-fire target and commence movement along a target attack route,

   c. commence movement to a defensive position to enhance survival,

   d. commence movement to a waiting position to await delivery of requested fire support,

   e. commence movement along a route that permits delivery of indirect MISTIC fire,

   f. commence movement along a route that permits illumination of targets being engaged with indirect MISTIC fire,

g. commence movement utilized during a search for targets, and

h. commence movement to rejoin the unit formation.

As discussed in a previous section, when a movement decision is formulated, the only processing required to indicate that the decision has been made is to record a value for the route selection parameter IRS to indicate that a new route for the section is required. However, because the section will be commencing a new route as a result of the decision, changes also occur in the section movement activity indicators JUNACT(NSEC) and JPHASE(NSEC). Moreover, it may be necessary to specify new values for the formation patterns and desired speed of the section. We will discuss these possibilities in the paragraphs which follow.

Decision Analysis

In this subsection, we will discuss in more detail the decision analysis performed according to the assumptions listed previously. The discussion is most conveniently conducted by analyzing each assumption separately. We use the following abbreviations for the movement activity indicators:

JUN = JUNACT(NSEC),
JPH = JPHASE(NSEC),
IUN = IUNACT(NAT), and
IPH = IPHASE(NAT).

Assumption One

A section conducting a firing run has JUN = 2 and JPH = 0. Since no decisions are allowed according to assumption one, the section continues its present activity. Therefore, JUN, JPH and the formation pattern and speed of the section remain unchanged. Moreover, the value of IRS that was set in the previous analysis in subroutine FLGSET remains unchanged since no superseding decision has been made. Finally, since the section has not been allowed to react to any pending unit decision, the section decision flag ISACT(ISEC), which was discussed earlier, remains unchanged. Of course, this means that the unit decision flag LMUFL(MANUN) cannot be reset either. See the description of subroutine HELFIR which appears in the discussion of mission area operations in a subsequent subsection for more information on this topic.

Assumption Two

A section that is retiring independently from the battlefield has JUN = 4. According to assumption two, the section must continue this activity. Therefore,

the processing that occurs in this situation is almost identical to that outlined for assumption one. The only exception is that the section is recorded as having reacted to any pending unit movement decision. This convention is followed since the only way retiring sections may react to a unit movement decision is to continue retiring. Therefore, if ISACT(ISEC) is positive, it is reset to zero and subroutine LMUSET is called to reset LMUFL(MANUN) if possible. Subroutine LMUSET was discussed in a previous section.

Assumption Three

According to assumption three, a section that is not firing or retiring independently must consider the decision to retire above all other decisions that are available. First, if the unit itself has decided to retire (IUN = 10) and the section has not reacted to the decision (ISACT(ISEC) > 0), then the section may react in one of two ways. If the section has been operating with the unit (JUN < 4), it will commence retirement as part of the unit formation. If the unit has been operating independently (JUN > 3), it will retire independently. In both cases, the section will be recorded as having reacted to the pending unit movement decision as indicated in the previous paragraph.

To commence retirement with the unit, the following processing is accomplished:

Maneuver Unit Leaders--Set	JUN = 0 and JPH = 0 to indicate that the section is in the unit enroute formation
Set	IRS = 6 so that a cross-country route will be selected
Other Section Leaders--Set	JUN = 0 and JPH = 1 to indicate that the section is joining the unit formation for enroute movement
Set	IRS = 1 so that a transition route will be selected for the section to join the unit
Set	the formation and speed of the section as required for enroute movement with the unit. This processing is accomplished in subroutine HFORM to be discussed subsequently.

To commence independent retirement movement, the following processing is accomplished:

Set JUN = 4 and JPH = 1 to indicate that independent retirement movement is under way.

Select a retirement position by calling subroutine DEFPOS. This subroutine has been discussed previously in Chapter 4.

Set the formation and speed of the section as required for independent retirement movement by calling subroutine SECPRM. This subroutine will be discussed subsequently.

Set IRS = 6 so that a cross-country route for the section will be selected.

Now, the discussion above was concerned with sections that are simply reacting to a unit retirement decision. However, it may be that the section decides to retire even though the unit has made no such decision. When this occurs, the section retires independently and the processing that must be accomplished is exactly the same as that outlined above for independently retiring sections.

The independent retirement decision is formulated in subroutine RETIRE. This subroutine returns the variable IRET which is defined as follows:

$$\text{IRET} \begin{cases} > 0 & \text{if the section should retire, and} \\ = 0 & \text{if otherwise.} \end{cases}$$

The rationale behind and the procedure used in subroutine RETIRE are given below.

An aerial vehicle section will retire from the battlefield when it can no longer function as an effective organization. The section is considered to be no longer effective if one or more of the following conditions exist for the current element's section.

1. The section's fuel supply, WFUEL(NSEC), is below the critical level, CFUEL(LWC), specified as input for aerial weapon code LWC.

2. The number of surviving elements in the aerial section is below the critical level, NREQR(LWC), specified as input for aerial weapon code LWC.

3. All elements of the aerial section have exhausted their ammunition supply.

The computational procedures to determine whether or not the aerial vehicle section to which the current element belongs should retire are given below.

1. Define ISEC = section number of the current element.

2. Set NSEC = NAVSEC(ISEC),
   IRET = 0, and
   IE = ISORG(1, ISEC).

3. If any elements remain in the section; i.e., IE > 0, go to step 5. Otherwise, continue.

4. Record the fact that no elements of the section are alive; i.e., set IRET = 2; go to step 21.

5. Determine LWC, the aerial vehicle weapon code for the section; i.e., set LWC = LWCOD(IE) - MAXLWC.

6. If the section's fuel supply is below critical level; i.e., WFUEL(LSEC) ≤ CFUEL(LWC), continue. Otherwise, go to step 8.

7. Record the fact that the section is to retire from the battlefield; i.e., set IRET = 1; go to step 21.

8. Set I = 1.

9. If the element in position I of ISEC is alive; i.e., ISORG(I, ISEC) > 0, continue. Otherwise, go to step 12.

10. I + 1 → I.

11. If I > 4, go to step 21. Otherwise, go to step 9.

12. If the number of survivors is below critical level; i.e., I < NREQR(LWC) - 1, go to step 7. Otherwise, continue.

13. Set I = 1.

14. Set IAM, the ammunition code, equal to 1.

15. If the element in position I is alive; i.e., ISORG(I, ISEC) > 0, continue. Otherwise, go to step 7.

16. Set IE = ISORG(I, ISEC).

17. Determine ICNT, the number of rounds of ammunition type, IAM, remaining for element IE; i.e., call AMMO(IE, IAM, ICNT).

18. If ICNT > 0, go to step 21. Otherwise, continue.

19. If IAM < 6, set IAM = IAM + 1 and go to step 17. Otherwise, continue.

20. If I < 4, set I = I + 1 and go to step 14. Otherwise, go to step 7.

21. The computations are complete.

Assumption Four

As stated in assumption four, if the retirement decision for a section is considered and then rejected, and if a unit movement decision is pending, then the decisions that are available to the section are only those that are consistent with the new unit movement policy. Thus, in this situation, a section always reacts to the unit decision. This reaction can be in the form of a continuation of movement already initiated in response to the unit decision or it can be in the form of a new section movement decision. In the latter case, the section has not previously reacted to the unit decision; i.e., ISACT(ISEC) > 0. However, after the decision is formulated, the variable ISACT(ISECT) is reset to indicate response and subroutine LMUSET is called to reset the unit decision flag LMUFL(MANUN) if possible.

When the section has not previously reacted to the unit decision (ISACT(ISEC) > 0), the section decision that is formulated is dependent upon both the present movement activity of the section and the movement activity of the unit. The decision analysis is as follows:

1. If the section is operating with the unit (JUN < 4), the section commences movement to permit the section to remain with the unit. Since all new unit movement decisions are such that the unit commences enroute movement, the section decision is to join the unit formation for this enroute movement. The processing required to implement this decision is displayed on page 6-39. There, the section was joining a unit for enroute movement off the battlefield.

2. If the section is operating defensively (JUN = 5), the section may continue this movement activity or it may commence movement to rejoin the unit. The first decision is formulated if the unit has commenced movement to the defensive position (IUN = 9). The idea here is that the section will simply wait for the unit to reach the defensive position and then join the unit formation. Thus, in this situation, no processing is required since no decision has been formulated. However, if the unit has not commenced movement to the defensive position (IUN ≠ 9), then the section decision is to rejoin the unit. The objective of the section becomes the objective of the unit (XS(NSEC) = OBJX(MANUN), YS(NSEC) = OBJY(MANUN)), the movement state variables are set to values that indicate enroute movement toward the unit (JUN = 6, JPH = 1), the route selection indicator is set so that a cross-country route will be selected (IRS = 6), and then subroutine DEFSET is called to zero out the recorded defensive position of the section if necessary. This subroutine will be described in a following paragraph. Finally, the speed and formation of the section are recorded for the enroute movement by subroutine SECPRM. This subroutine will also be described subsequently.

3. If the section is already enroute to join the unit (JUN = 6) when the unit decision is made, the section will continue toward the unit. However, the final destination of the section changes to allow the section to move toward the new unit mission area. This means that the route must also change. The processing required is almost exactly the same as that outlined for a section commencing movement from a defensive position in situation 2 above. The exceptions are that subroutine DEFSET need not be called, and JUN need not be set since it already has the proper value.

Assumption Five

According to assumption five (page 6-37), in the absence of any preempting decisions that arise because of decisions made according to assumptions one through four, a section formulates decisions that are consistent with the present movement state of the section. The decisions that are available are listed on pages 6-37 and 6-38.

The decision analysis that precedes the analysis associated with assumption five is such that the only sections falling under the decision rules of assumption five are those that are operating with the unit (JUN < 4). The reader may verify this statement by carefully analyzing the discussion on preceding pages. Thus, the sections of interest are:

1. traveling with the unit in an enroute formation (JUN = 0);

2. searching for targets either independently or in the unit formation, or loitering in the unit formation (JUN = 1);

3. conducting an illumination or indirect-fire MISTIC firing assignment, or conducting a direct-fire attack (JUN = 2); or

4. waiting for fire support to be delivered against a target located by the section (JUN = 3).

Of these four activities, only the latter three are of particular interest. If the section is performing the first activity, it is not allowed any movement decisions or target assignments. The section must simply continue movement with the unit formation, and no processing is required in subroutine HELCON. This operating policy is based on the TAPCOM II assumption stated in Chapter 1 that specifies that the sections of a unit are allowed to engage the enemy only when the unit is in its mission operations area.[1]

Even in the latter three situations cited above (JUN > 0), the unit must actually be performing a mission before any activity decisions of the sections

---

[1]The reader will recall from Chapters 1 and 4 that should an aerial unit encounter an enemy threat suitable for attack while performing enroute movement, it must abort its present mission and begin a self-defense mission before attacks can be performed. If this alternative is selected, then the unit will, by definition, be in the operations area of the self-defense mission so that attacks can be conducted (see Chapter 4).

are allowed. This means that MCLASS(NAT) must be greater than one (see Table 6.4), or alternatively, $0 < IUN < 9$. If the unit is loitering at a defensive position (IUN = 9), waiting for a new mission (IUN = 0) or has retired from the battlefield (IUN = 10), sections operating with the unit have JUN = 1 and have no battlefield responsibilities against the enemy. They are allowed only to continue their loitering activities. Therefore, no processing is required in subroutine HELCON when JUN = 1 and MCLASS(NAT) = 1.

With the exceptions cited above, processing is required in subroutine HELCON to determine movement decisions for the section and to implement these decisions. The situations that remain involve sections that are operating with the unit ($0 < JUN < 4$) in the mission operations area, and the sections have battlefield responsibilities.

Mission Area Operations

The decision analysis to be discussed for mission area operations is performed in subroutine HELFIR whose flow chart appears in Volume 2. The analysis is summarized in Table 6.10. Here, the available decisions are listed according to the mission of the unit and the current activity state of the section. Notice that in all cases, a continuation of the present activity is the default decision.

Now, as specified previously, the variable IRS must be set any time that a movement decision is formulated, so that the route selection model will know the type of route to be constructed. The values of IRS that are used and their meanings are indicated in Table 6.6. Moreover, the section state variables JUN and JPH change to reflect the new movement activity and status. Finally, the section speed and formation pattern must be selected to be in agreement with the new movement activity. The reader will recall that subroutine SECPRM conducts the required processing when the section is moving independently of the unit, while subroutine HFORM is used when the section is considered a part of the unit formation. Both subroutines have a calling parameter IFTN that indicates the type of movement for which speeds and formations are desired. The values are:

$$\text{IFTN} = \begin{cases} 1 & \text{for a cross-country route,} \\ 2 & \text{for a loiter station route,} \\ 3 & \text{for a search route, and} \\ 4 & \text{for an attack route.} \end{cases}$$

Both subroutines will be discussed in more detail in a later paragraph.

Table 6.10

Mission Area Decision Analysis

Unit Mission	Section Movement Activity	Available Movement Decision
Direct-Fire Attack (MCLASS(NAT) = 2)	Flying search route (JUN = 1)	Continue Commence attack route Commence cross-country route
	Flying attack route (JUN = 2)	Continue Commence search route Commence different attack route Commence cross-country route
Indirect-Fire Attack (MCLASS(NAT) = 4)	Flying loiter station route (JUN = 1)	Continue Commence attack route
	Flying attack route (JUN = 2)	Continue Commence route to reenter loiter station
	Flying route to reenter loiter station (JUN = 1)	Continue Commence attack route
MISTIC Forward Observer (IUN = 6)	Flying search route (JUN = 1)	Continue Commence route to waiting position Commence illumination route
	Flying at or to waiting position (JUN = 3)	Continue Commence search route Commence illumination route
	Flying illumination route (JUN = 2)	Continue Commence search route Commence route to waiting position
Counterbattery Observation (IUN = 7)	Flying search route (JUN = 1)	Continue
Special Forward Observer (IUN = 8)	Flying search route (JUN = 1)	Continue Commence route to waiting position
	Flying at or to waiting position (JUN = 3)	Continue Commence search route

Table 6.11 has been prepared to indicate the processing that is required when each of the decisions indicated in Table 6.10 are formulated. The table indicates the new values of IRS, JUN, JPH and IFTN. The table also indicates whether subroutine HFORM or subroutine SECPRM is used to establish speed and formation patterns. The only processing that must be accomplished to implement decisions, that is not shown in Table 6.11, occurs when a decision is made to seek a defensive position or a waiting position. When these decisions are formulated, subroutine DEFPOS must be used to determine the location of the position. This subroutine was discussed in detail in Chapter 4.

The decision analysis presented in Table 6.10 becomes more understandable as we discuss in more detail the activities that are simulated for sections during mission area operations. The topics in the order they will be discussed are:

1. direct-fire activities,

2. indirect-fire activities,

3. forward observer activities, and

4. counterbattery observation activities.

Direct-Fire Activities

When a section enters a mission operations area of a direct-fire mission (MCLASS(NAT) = 2), the section commences a route to be used while searching for targets to be attacked. The processing required to initiate this activity is conducted in subroutine FLGSET discussed previously. In the absence of preempting decisions discussed earlier, search movement continues until a suitable target is located or until the section decides to seek a defensive position. In the former case, an attack route is selected while in the latter case, a cross-country route to the defensive position is selected. The choice between continued search, attack and defense is formulated either in subroutine CBFIR (for counterbattery attack missions) or in subroutine AIRFIR (for all other direct-fire missions). Both these subroutines operate on the premise that the decision to attack is made only when a suitable target complex consisting of one or more enemy elements has been located. For counterbattery fire, the complex must consist of components located at the assigned enemy artillery battery while for all other direct-fire missions, the complex must consist of regular vehicular enemy ground elements. However, the requirements stated above form only a framework within which the target selection schemes may operate. The selected complex must contain at least one enemy element of value. If such is not the case, the section should continue search movement. On the

Table 6.11

Implementing Movement Decisions

				Subroutine Used		
Decision	IRS	JUN	JPH	IFTN	SECPRM	HFORM
Commence attack or illumination route	7	2	1	4	x	
Commence search route	2	1	0	3	x	
Commence different attack route	10	2	1	4	x	
Commence cross-country route						
a. for defense	6	5	1	1	x	
b. in response to unit decision	6	0	1	1		x
c. for retirement	6	4	1	1	x	
Commence route to reenter loiter station	1	1	1	2		x
Commence route to waiting position	6	3	1	1	x	

other hand, if no complex exists that can be attacked without undue risk to the aerial section, the section should seek a defensive position as opposed to continuing search movement or attacking a target.

Subroutine CBFIR or subroutine AIRFIR is called in subroutine HELFIR each event of the section leader when the section is searching for direct-fire targets. As will be discussed, subroutine AIRFIR is also called in some cases at the end of a target attack. If no target is selected, search continues and no decision implementation is required. If defense or attack is selected, then the processing indicated in Table 6.11 is performed.

Now, when an attack is decided upon in subroutine AIRFIR, processing is also accomplished internally to assign targets to elements of the section. These assignments are made prior to initiation of attack movement and are valid for the duration of the attack. The assignments specify which weapon or weapons are to be fired by each element in the section, and at which enemy element or elements. Details of the model are presented in the target selection model discussion that appears at the end of this chapter.

Once the decision has been made to conduct a direct-fire attack, the section commences movement along the attack route. To understand the movement decisions that may be made subsequent to the initiation of this movement, we must fully understand the activities that are conducted during the attack.

As discussed in the route selection model section of this chapter, a direct-fire attack consists of three segments. During the first segment, the section is moving toward a point called the initial point (IP) from which firing may commence. Target assignments already exist since they were made at the time that the decision to attack was made.

The second segment is called attack phase one, and for direct-fire attacks, will occupy exactly one event of each element in the section. During attack phase one, a helicopter in the section may fire according to one of the following doctrines:

1. no firing,

2. multiple bursts from one suppressive fire weapon at one target,

3. multiple bursts from two suppressive fire weapons at either one or two targets,

4. single shot from a point-fire destructive weapon, or

5. single shot from a point-fire destructive weapon preceded by multiple bursts from a single suppressive fire weapon. Both weapons fired at same target.

The third segment exists if and only if a point-fire destructive weapon was fired by some element of the section during attack phase one. This segment is called attack phase two and is included only to allow suppressive fire to be delivered after the delivery of point destructive fire. The segment terminates as soon as all point-fire weapons have impacted. However, it may be that the segment continues for several events. This situation can occur if a MISTIC missile is fired in the direct-fire mode since the missile is a separate element that takes an indeterminant number of firer events to fly and impact. For all other projectiles, the flight time can be computed during the event of the firer and used in determining his event time.

During attack phase two, a helicopter in the section may fire according to one of the following doctrines:

1. no firing,

2. multiple bursts from one suppressive fire weapon at one target, or

3. multiple bursts from two suppressive fire weapons at either one or two targets. This case can occur only as a continuation of the similar case of phase one and involves the same targets and weapons.

The attack may be terminated at the end of attack phase one only if no point-fire weapon was fired during attack phase one as discussed above.

If phase two does commence, then it may terminate after only one event of the section or after several events. The first case arises if no element of the section fires a direct-fire MISTIC missile. The latter case occurs if at least one MISTIC missile is fired. The reasoning behind these statements was presented earlier.

At completion of an attack, at the end of either phase one or phase two, a reevaluation of the tactical situation is performed. The decision analysis is performed according to the following scheme:

1. If the section should retire independently, then the section will commence independent retirement movement. Otherwise, continue.

2. If a movement decision has been formulated by the unit during the attack conducted by the section, then the section will react to the decision. Otherwise, continue.

3. If the section should seek a defensive position, then the section will commence enroute movement to a defensive position. Otherwise, continue.

4. If the section should reattack the target that was just attacked, then the section should commence movement to the new initial point. Otherwise, continue.

5. The section should commence movement to allow a search for additional targets.

Now that we have discussed the principles of an attack and the decisions that may be made, we may discuss the variables used in the analysis.

As stated previously, either subroutine AIRFIR or subroutine CBFIR is called to determine whether the section should commence an attack, seek a defensive position, or continue searching. To indicate the results of the analysis, both subroutines return the variable ITGASN defined as follows:

$$\text{ITGASN} = \begin{cases} 0 & \text{if section should seek a defensive position,} \\ 1 & \text{if section should continue searching, and} \\ 2 & \text{if section should commence an attack.} \end{cases}$$

In an attack is indicated (ITGASN = 2), then the target assignments are prepared in subroutine AIRFIR[1] and are indicated by two arrays. These

---

[1]As presently formulated, TAPCOM II includes only a dummy version of subroutine CBFIR. This subroutine always returns ITGASN = 1. Thus, aerial units conducting a counterbattery attack mission will never attack the assigned battery. The reason for this feature of the model is that in general an aerial unit would have to leave the battlefield to conduct attacks on an artillery battery. In TAPCOM II, such operations are not allowed. If at some

arrays reveal not only the enemy elements to be attacked by the elements of the section, but also the ammunition to be used. The arrays are as follows:

IHTARG(I, LHCE) = target element of helicopter element LHCE.

IHAMO(I, LHCE) = ammunition code of weapon to be used by helicopter element LHCE

where

$$I = \begin{cases} 1 \text{ indicates weapon one, phase one,} \\ 2 \text{ indicates weapon one, phase two,} \\ 3 \text{ indicates weapon two, phase one,} \\ 4 \text{ indicates weapon two, phase two, and} \\ 5 \text{ indicates point-fire destructive weapon.} \end{cases}$$

These two arrays are filled by subroutine CONVRT for all helicopters LHCE in section NSEC. Subroutine CONVRT is called at the very end of processing in subroutine AIRFIR and operates on targeting data prepared by subroutine WASAIR. The reader will note in the above definitions that, for convenience, suppressive fire weapons are referred to as weapon one and weapon two since as many as two may be fired during an attack phase (see previous discussion of attacks). No significance is attached to this convention, however.

Now, as reported in Chapter 8, the aerial firing model is designed to use presently existing ballistic weapon lethality models and missile flight models. Thus, the targeting data prepared by subroutine AIRFIR must be converted into a form that is consistent with the data requirements of these existing models. The conversion process occurs in subroutine ATKPRM.

The variables that must be prepared for either phase one or phase two attacks are as follows:

---

future time, the user desires to incorporate features that allow aerial units to leave the battlefield while performing a mission, or if artillery units are placed on the battlefield, then subroutine CBFIR could be developed using an approach that is similar to that of subroutine AIRFIR.

MDFAF(J)	=	the element number of the first target to be attacked by element J in section ISEC (if any)
LTARG(J)	=	the element number of the second target to be attacked by element J in section ISEC (if any)
LFRND(J)	=	first round flag for element J in section ISEC (associated with target MDFAF(J))
KFRND(J)	=	first round flag for element J in section ISEC (associated with target LTARG(J))
LRNDC(J)	=	round count for element J in section ISEC
TFLY(L)	=	launch preparation time for helicopter L in section NSEC
LNFIR(LTG)	=	the number of firers engaging enemy element LTG.

We will discuss each of these variables in more detail in the paragraphs that follow.

The variables MDFAF(J) and LTARG(J) indicate the targets of each element J in the aerial section. Table 6.12 summarizes the convention that is used for the assignment of values based upon the attack phase and firing doctrine being employed. The convention is used only for associative purposes since two targets may be assigned to each aerial element.

The first round flags LFRND(J) and KFRND(J) are set to one as needed at the beginning of attack phase one and are reset to zero as soon as the weapons with which they are associated are fired. These variables are used in the firing model to differentiate between ballistic weapon accuracy data sets that are dependent upon whether or not a particular weapon has been fired. At the beginning of attack phase two, the first round flags are again set to one only if new weapons, not fired during phase one, are to be fired.

The round count indicator LRNDC(J) is used throughout DYNCOM to indicate the remaining rounds to be fired by element J at its existing target. When an assignment is made, the round count is initialized from input data and thereafter is decremented by one each time a round is fired. In TAPCOM II, the round count only indicates whether or not element J is to fire during a particular attack phase. Thus, at the beginning of phase one and again at the beginning of phase two, LRNDC(J) is set to one if and only if element J is to fire during the phase. Otherwise, a value of zero is entered.

Table 6.12

Target Data Associations

Weapon	Phase	Subroutine AIRFIR Processing Results					
		1	2	3	4	5	6
Suppressive Fire	1	$\boxed{X}$	0	$\boxed{X}$	0	X	X
Weapon One	2	0	0	0	0	X	X
Suppressive Fire	1	0	0	0	0	0	$\boxed{X}$
Weapon Two	2	X	X	0	0	0	$\boxed{X}$
Point-Fire Weapon	1	X	X	X	X	0	0

X indicates firing by designated weapon to occur during designated attack phase

0 indicates no firing

○ indicates target of designated weapon to be specified as MDFAF(J) during designated attack phase

□ indicates target of designated weapon to be specified as LTARG(J) during designated attack phase

Example: Subroutine AIRFIR Processing Result One

Phase I: MDFAF(J) = IHTARG(5, L)
LTARG(J) = IHTARG(1, L)

Phase II: MDFAF(J) = IHTARG(4, L)
LTARG(J) = 0

The variable TFLY(L) is computed in subroutine TFCOMP (or ATKPRM for indirect-fire MISTIC missiles) at the time of target assignment. The value recorded is always zero unless element J is to fire a point-fire destructive weapon. Otherwise, the value represents the amount of time that is required after cessation of suppressive fire for element J to actually launch the point-fire projectile. The relationship is

$$TFLY(L) = THBM(K, LCOD) + N(0,1) * SHBM(K, LCOD)$$

where

(THBM(K, LCOD) and SHBM(K, LCOD) are the mean and standard deviation, respectively of the distribution of times required to fire a point-fire weapon of type K from a helicopter having aerial weapon code LCOD, and

N(0, 1) is a sample value from a normal distribution with zero mean and unit variance.

The reader will recall that LCOD is found from the relation

$$LCOD = LWCOD(J) - MAXLWC$$

where MAXLWC is as defined in common area /NUMBER/.

The subscript K is defined as follows:

$$K = \begin{cases} 1 & \text{if a MISTIC missile is being launched,} \\ 2 & \text{if a beam rider missile is being launched, and} \\ 3 & \text{if a point-fire ballistic weapon is being fired.} \end{cases}$$

To determine the proper value of K, the input array IHDFMC is used in conjunction with the point-fire ammunition assignment variable IHAMO(5,J) that was discussed earlier.

The array IHDFMC specifies the types of point-fire ammunition that are carried by an aerial vehicle; i.e.,

IHDFMC(I, LCOD) = ammunition code for point-fire weapon type I being carried by a helicopter with aerial weapon code LCOD

where

$$I = \begin{cases} 1 & \text{for MISTIC missile, and} \\ 2 & \text{for beam rider missile.} \end{cases}$$

Then, K is found by the following logical procedure:

1. Set K = 1.

2. If IHAMO(5, L) = IHDFMC(1, LCOD), go to step 8. Otherwise, continue.

3. Set K = 2.

4. If IHAMO(5, L) = IHDFMC(2, LCOD), go to step 8. Otherwise, continue.

5. Set K = 3.

6. If IHAMO(5, L) > 0, go to step 8. Otherwise, continue.

7. No point-fire assignment exists.

8. The procedure is complete.

The variable TFLY(L) is reset to zero in the firing model if a MISTIC missile is launched. If a MISTIC missile was not launched, it is set to a value that represents the flight time of the beam rider missile or ballistic projectile that was fired (if one was fired). Then, at the end of attack phase one, TFLY(L) can be used to determine whether or not attack phase two is to be conducted and if so, the amount of time to be used as the event time for phase two. We will discuss this process in more detail in a subsequent paragraph.

The variable LNFIR(LTG) represents the total number of firers on enemy element LTG and is used primarily in the ground weapon fire controller during target assignment procedures. When helicopters attack targets, it is assumed that ground weapons are aware that the attack is underway. Thus, when aerial element J is assigned a target LTG, the variable LNFIR(LTG) must be incremented by one. The procedure accounts for the fact that as many as two targets may be assigned to each element in the aerial section. On the other hand, each element may fire as many as two weapons at the same target. In the first case, both targets have their firer number incremented while in the second case, the firer number is incremented only by one.

At the end of phase one and again at the beginning of each event of phase two, it must be determined whether or not the attack has been completed. The reader will recall that the conditions for completion are:

1. no point-fire weapons were assigned during phase one; or

2. all point-fire weapons fired during attack phase one have impacted.

If condition one is satisfied, attack phase two will never commence. Otherwise, attack phase two will occupy one or more subsequent events to allow for flight time of point-fire projectiles.

To determine whether condition one above is satisfied at the end of attack phase one, the point-fire target assignment data IHTARG(5, L) for all helicopters L in section NSEC are used. Condition one is satisfied and attack phase two does not start if and only if

$$\sum_{L \in NSEC} \text{IHTARG}(5, L) = 0.$$

If condition one is not satisfied, then on each event of phase two, it is determined whether condition two is satisfied. Each helicopter L in section NSEC is analyzed to determine whether it launched a point-fire weapon during phase one, and if so, whether the projectile has impacted. For each helicopter, the following analysis is performed.

1. If TFLY(L) > 0, helicopter L launched a beam rider missile or ballistic weapon during phase one and this is the first event of phase two. Set the event time TDEL to TFLY(L) and go to step 5. (The reader will recall that TFLY(L) is set to a positive value in the firing model if and only if a ballistic weapon or beam rider missile was launched.)

2. If TFLY(L) ≤ 0, helicopter L either launched a MISTIC missile or fired no point fire weapon. If IHTARG(5, L) > 0 and IHAMO(5, L) = IHDFMC(1, LCOD), a MISTIC missile was fired. Go to step 3. Otherwise, compute the event time TDEL = 0 and go to step 5.

3. If the missile has impacted, compute the event time based on the impact time and go to step 5. Otherwise, continue.

4. Compute the event time as a standard amount of time to be used while the missile is flying; i.e., set TDEL = EVHTIM(LWC) where the array EVHTIM is input.

5. Processing is complete.

At step 3 above, to determine whether or not the missile launched by helicopter L is still flying, we use the variable LFLAG(L). This variable is set to one in the helicopter launch model of Chapter 8 when a missile is launched and is reset to zero in the missile flight model when the missile impacts or aborts. If LFLAG(L) = 0, the missile has impacted and the time

of impact is recorded as TDFRDY(L). This latter variable is set in the missile flight model when LFLAG(L) is reset at impact. The event time then is TDEL = TDFRDY(L) - ECLOCK(J) where ECLOCK(J) is the present clock time of element J (corresponding to helicopter L).

To determine whether condition two is satisfied at any event, we construct the event time

$$\text{DELT} = \max_{L \in \text{NSEC}} \text{TDEL}$$

where TDEL is the value for each helicopter L as constructed by the procedure outlined above. If DELT = 0, condition two is satisfied and attack phase two can terminate. However, if DELT > 0, it may be used as the event time for each element in the section. Thus, TIME = DELT where TIME is the event time recorded in common area /ICECOM/ for the current element.

Now, whenever condition one or condition two are satisfied, the attack can terminate. Thus, we perform the decision analysis that was indicated earlier to decide what the next activity will be. First, we determine whether the section should retire by calling subroutine RETIRE that was described earlier. If the section should retire, parameters for independent retirement movement are established as indicated in Table 6.11.

If retirement is not indicated, then we determine whether a unit movement decision is pending that the section has not reacted to. The reader will recall that this can be determined by the conditions LMUFL(MANUN) > 0 and ISACT(ISEC) > 0. If these conditions are obtained, then the section reacts by commencing movement to join the unit formation (see Table 6.11).

If the section still has not made a decision after the above two steps, subroutine AIRFIR is called to determine whether the section should again attack the target, revert to search for other targets or seek a defensive position. Processing at this point is almost exactly the same as that described earlier when an initial target attack was being considered. The only difference is that a value of ten for the route selection parameter IRS is set in the event that a reattack is desired. This value indicates that a route is to be selected according to rules for subsequent attacks by the route selection model. See Table 6.11 for implementation data to be used for the defense, search and reattack decisions.

Indirect-Fire Activities

Units performing the indirect-fire MISTIC launcher mission commence operations area activities by occupying a loiter station. A section of the unit continues occupying this loiter station as part of the unit formation until it

receives an indirect-fire assignment. Then, it commences movement that will allow the launching of missiles in the direction of the target. Upon completion of the assignment, the section commences movement back to the station where it again reenters the unit formation. All of these activity decisions are indicated in Table 6.10.

Now, in subroutine HELFIR, the movement and firing decisions associated with sections of a unit performing the indirect-fire MISTIC mission need be analyzed. Target assignment and communication activities of the section are processed by the indirect-fire MISTIC launcher fire-support model described in Chapter 3. That model is implemented in subroutine MFB and simulates all the communication required during mission assignment and termination. The element that is processed in subroutine MFB is the launcher element NUM that is associated with the aerial section NSEC. The reader will recall that this association is established in subroutine NEWFO which was described earlier. The launcher fire-support element is activated at the time that the section enters the operations area and remains active until the section leaves the operations area.

To determine what movement decisions should be made by NSEC at any time, the mission phase indicator IFBMIS(NMF) is used, where NMF is the fire-support firer number of element NUM (NMF = NUM + NUMART). The definition of IFBMIS(NMF) is as follows:

$$\text{IFBMIS(NMF)} = \begin{cases} 0 & \text{NUM has no mission,} \\ 1 & \text{NUM has a mission that can be fired as soon as a firing position is achieved (confirmed mission), and} \\ 2 & \text{NUM is currently waiting for mission confirmation.} \end{cases}$$

MFB searches the list of MISTIC fire requests submitted by FO's for the oldest request. Having selected a fire request, MFB sets IFBMIS(NMF) to 2 and initiates a target verification message with the FO. Once the verification message is received by the FO, subroutine AFO verifies the target existence, sets IFBMIS(NMF) to one, and prepares the following firing data for subsequent use:

IHAMO(J, L) = IHDFMC(1, LCOD)
MDFAF(J) = IPICK
IHTARG(J, L) = IPICK

where IPICK is the highest priority target element number selected by subroutine AFO.

Within subroutine HELCON, the convention is used that unless the mission is confirmed (IFBMIS(NMF) = 1), the section should be occupying the loiter station or at least attempting to enter the loiter station formation. Only when a mission is confirmed should the section be moving toward a fire position (by convention, along an attack route).

Thus, we have the following decision analysis for indirect-fire sections:

1. If NUM moved into a fire position during its last event (MSFP(ICE) = 1) and has a confirmed indirect-fire mission (IFBMIS(NMF) = 1) set IFIR and KFIREV to one so that subroutine HFIRE (see Chapter 8) will launch a MISTIC missile this event; go to step 5. Otherwise, continue.

2. If NUM has a confirmed mission (IFBMIS(NMF) = 1) and the section is moving toward a firing position (JUNACT(NSEC) = 2), continue the present activity; go to step 6. Otherwise, continue.

3. If NUM has a confirmed mission but the section has not commenced movement along an attack route (JUNACT(NSEC) ≠ 2), the section should commence attack movement; go to step 6. Otherwise, continue.

4. If NUM does not have a confirmed mission (IFBMIS(NMF) ≠ 1) and the section is in, or moving toward, the loiter station (JUNACT(NSEC) = 1), continue the present activity; subroutine ATKPRM is called to Monte Carlo for a value of TFLY(L); go to step 6. Otherwise, continue.

5. If NUM does not have a confirmed mission and the section is not moving toward the loiter station (JUNACT(NSEC) ≠ 1), the section should commence movement toward the station.

6. The decision analysis is complete.

Implementation of the decisions analyzed above is performed by the processing indicated in Table 6.11. The reader will note that to commence an attack route the same processing is used for indirect-fire sections as is used for direct-fire sections. The route selection model to be reported is designed to recognize the route requirement differences between the two types of sections. Moreover, the subroutine HELFIR is designed to allow an indefinite number of events to transpire during phase one of an attack being conducted by an indirect-fire section. Because of the way that indirect-fire attack routes are structured, it is assumed that an indirect-fire section will never reach the end of an attack phase one route and consequently, attack phase two will never commence. Instead, it is assumed that the mission will end before the recorded route has been entirely traversed.

Forward Observer Activities

Sections belonging to a unit conducting either a MISTIC or a special forward observer mission commence mission area operations by conducting a search for targets. The sections always operate independently.

Once a suitable target complex has been located by a section, a request for fire support (MISTIC, artillery or air) is prepared and sent. The section then stands by for the requested fire to be delivered. If MISTIC or aerial support has been requested, the section will be asked to verify the target prior to delivery of fire. If artillery fire has been requested, the section may be asked to observe the fire delivered, and to transmit fire adjustment messages. Finally, if MISTIC fire has been requested, the section must illuminate a target in the target complex for each missile fired.

Movement conducted by the section depends both upon the type of mission being flown and the present state of the section with respect to the fire-support activities indicated above. If the section is searching for targets, then a route is flown that permits effective search for targets. As soon as a target complex is located, the section commences movement to a waiting position at which the section may loiter while waiting for fire to be delivered. This position will be occupied for the duration of the mission unless MISTIC fire has been requested. In this event, the section commences movement along an attack route that permits illumination of targets as soon as missile fire commences.

Only the movement decisions indicated in the previous paragraph need be analyzed in subroutine HELFIR since all the other fire-support activities discussed are simulated in subroutine AFO (reference 1). This subroutine processes forward observer element NUM with which section NSEC is associated. The reader will recall that element NUM is activated by subroutine NEWFO when section NSEC enters the operations area and deactivates the element when NSEC leaves the operations area.

The movement decision analysis is based upon the forward observer state variables KFOD(NUM) and MISFO(NUM). Within subroutine HELCON, these variables have the following operational definitions.

1. KFOD(NUM) = 0 indicates that NUM has no target and consequently, that NSEC should be flying a search route.

2. KFOD(NUM) = 3 and MISFO(NUM) = 3 indicates that NUM has decided to cancel a mission that has been requested. Consequently, section NSEC should be flying a search route for a new target complex.

3. KFOD(NUM) = 2 and MISFO(NUM) = 1 indicates that NUM has been asked to verify the target and has responded with a mission confirmation. If NSEC is performing a MISTIC mission and MISTIC fire has been requested by NUM, then NSEC should be flying an attack route.

4. In all other cases, section NSEC should be occupying a waiting position since NUM has a target but no action by NUM is required.

To implement any decision made after analyzing KFOD(NUM), MISFO(NUM) and JUNACT(NSEC), the processing indicated in Table 6.11 is used. The reader will again note that an attack decision for MISTIC FO sections is implemented exactly in the same way as the attack decision for a direct-fire section. However, the same observations that were made in the discussion of indirect-fire attack decisions apply in this case as well.

Counterbattery Observation Activities

As indicated in Table 6.10, sections of a unit performing a counter-battery observation mission have no movement decisions available to them. Upon entry into the mission operations area, the sections perform a search for the assigned enemy artillery unit and continue this activity until the mission is terminated. Moreover, all the sections of the unit are constrained to move with the unit formation while conducting the search.

Some processing is required in subroutine HELFIR, however. When the section leader is also the maneuver unit leader, subroutine CBOBS is called to determine whether or not the unit has located the assigned enemy artillery unit. If so, the time at which the detection occurred is recorded for later use when the mission is terminated (see Chapter 4).

For counterbattery observation missions, the fact that the artillery unit has not been located by NAT is indicated by NVOLM(NF) > 0 where NF is the fire-support firer number of NAT. If NVOLM(NF) ≤ 0, the battery has already been located and no processing is required in subroutine CBOBS.

Now, if observation has not occurred (NVOLM(NF) > 0), it must be determined whether observation took place during the last event. The duration of the last event is TIM = ETIM(ICE) where ICE is the element number of the current element.

The model used to predict whether or not detection occurred during TIM is based upon the detection rate concept that has been used in other DYNCOM

detection models. These models assume that for small time intervals the rate of detection λ is constant, and this approach yields the expression:

$$p = 1 - e^{-\lambda \Delta t}$$

for the probability of detection p during the time interval Δt.

The difference between the counterbattery observation model and other DYNCOM detection models is that in the other models the detection rate λ has been estimated from laboratory and field data for a variety of tactical situations, and these estimates have been incorporated into the detection models. However, the incorporated estimates are valid only for regular target elements such as tanks and APC's. No estimates exist for area targets such as an artillery unit nor for that matter, the componenets of these targets such as artillery tubes or communications facilities. Thus, λ must be prepared by the user and input to the simulation.

The detection rate for an aerial unit with weapon code LWC is CBDET(LWC). The aerial unit weapon code is defined as in the introduction to this chapter and is used as the subscript in the array CBDET because it is assumed to be an indicator of the detection capabilities of the unit. However, CBDET is not structured to reflect differences between types of enemy artillery units or observer scene variables such as range. If this feature were desired by the user, additional subscripts could be introduced to indicate the desired dependency.

The processing of subroutine CBOBS is as follows:

1. Determine the probability of successful observation; i.e., compute

   POBS = 1 - EXP(- CBDET(LWC) * TIM).

2. Obtain a random number R from a uniform distribution defined on the interval (0, 1).

3. If R > POBS, detection did not occur. However, if R ≤ POBS, detection did occur. Therefore, set NVOLM(NF) = 0 to indicate successful observation. Also, record the time at which detection occurred; i.e., set TMISUN(NAT) = ECLOCK(ICE) where ECLOCK(ICE) is the current clock time of the current element.

Selecting and Recording Routes

As noted in the introduction of this chapter, ten distinct route selection processes exist in TAPCOM II. Procedures required to implement these route selection processes and to record the selected routes are accomplished in step 8 of the processing sequence of subroutine HELCON. All procedures are performed in subroutine PICKRT.

The parameter that indicates the type of route to be selected is IRS, and the values permitted are as shown in Table 6.6. We will discuss the processing that is accomplished for each value after we have given an introduction to the general structure of the model and have defined the principal variables that are used.

The general processing scheme is as follows:

1. Depending upon IRS, choose a subroutine to select the type of route desired. The subroutine selected constructs and returns a description of the route in the form of three working arrays.

2. Call subroutine HXYMCP to record the selected route in common areas that contain a description of the route being flown by the section and/or the unit.

The working arrays prepared at step 1 of the above sequence are XOPT, YOPT, ZOPT. These arrays give the X, Y and Z coordinates, respectively, of the sequence of points that describe the new route segment for the section. The entries are ordered so that the first set of values always corresponds to the present position coordinates of the section leader. The last set of values represents the position of the end point of the route segment. Z is measured relative to the DYNCOM terrain zero elevation plane.

The common areas that contain the route descriptions for sections are /HXRT/, /HYRT/, and /HZRT/. The arrays contained in these three areas are identical in structure to one another and are loaded from XOPT, YOPT, and ZOPT, respectively. For example,

HXRT(I, J) = X coordinate of the I^{th} point in the route for aerial section number J where I = 1,...,IRTSIZ and J = 1,...,NASEC.

The variable IRTSIZ is contained in common area /RTSIZE/ and represents the maximum number of points in a section route. The variable NASEC represents the number of aerial sections being represented.

The reader should note that the route is actually loaded in an order that is reversed from that used in constructing the arrays XOPT, YOPT, and ZOPT. That is, the point corresponding to I = 1 is actually the last route point that will be encountered by an aerial section J.

The common areas that contain route descriptions for aerial units are /XRT/, /YRT/, and /ZRT/. These common areas are the same ones that are used for other types of maneuver units. The arrays contained in the three common areas are identical to one another in structure, and for aerial units, contain the route arrays that describe the route being flown by the section leader that is actually leading the aerial unit formation. This topic will be discussed in more detail in a subsequent paragraph.

An example of the structure used in the route arrays is provided by the definition of XRT; viz.,

XRT(I, J) = X coordinate of the I^{th} point in the route for maneuver unit J, where I = 1, . . . , IRTSIZ and J = 1, . . . , MNMNU.

The variable MNMNU is the number of maneuver units as defined in common area /NUMBER/.

The reader should note that the same ordering of route points is assumed in XRT, YRT, and ZRT as is used in HXRT, HYRT, and HZRT. Also, the maneuver unit number corresponding to aerial unit NAT must be determined before the arrays XRT, YRT, and ZRT may be used. Recall that MANUN = KMANU(NAT).

The length of the six arrays defined above is IRTSIZ as indicated. However, the number of valid points in the arrays varies from one section to another and changes each time a section selects a route. Thus, the valid lengths must be recorded. The array ICAP1 contains the valid lengths of route arrays for maneuver units and is arranged ICAP1(I); I = 1, . . . , MNMNU.

The array ICAP2 contains the lengths of route arrays for aerial sections and is arranged ICAP2(I); I = 1, . . . , NASEC.

Knowledge of the position of an aerial element along its route is maintained at all times by using the array LCPE. This array is used in other

portions of DYNCOM but in TAPCOM II is defined as follows:

LCPE(I) = the point in the arrays HXRT, HYRT, and HZRT to which element I is currently proceeding, where I = 1, . . . , NUMELE and NUMELE equals the number of vehicular elements as defined in common area /NUMBER/.

Note that the element number as opposed to the helicopter number of the aerial element is used in LCPE.

Now that we have given an overview of the route selection processing that is accomplished in subroutine PICKRT, we will discuss the various procedures that are used in more detail. The first discussion is concerned with the recording of a selected route while subsequent discussions will describe the various route selection procedures.

Recording a Route

As indicated previously, the route selected by one of the procedures to be discussed is described by the working arrays XOPT, YOPT, and ZOPT. The number of valid points in the arrays is indicated by IKAP which is contained in common area /ICAP/. The task of loading the working array information into the permanent route arrays for a section and/or a unit is performed in subroutine HXYMCP.

The task that must be accomplished in subroutine HXYMCP is illustrated by the example in Figure 6.1. Here, the "old" route consisted of seven points (ICAP2(N) = 7) and the current element was on the fifth route segment from the end (LCPE(ICE) = 5). A new route is chosen that has four valid points (IKAP = 4) starting with the current element's present position. The "new" route description prepared by subroutine HXYMCP consists of six points (ICAP2(N) = 6), two points from the "old" route that have already been traversed and four points from the "new" route that remain to be traversed. Note that the procedure saves as much of the "old" route as possible for use by other elements that may be guiding on the section leader ICE.

From the above illustration, we see that a new route for a section is recorded by performing the following tasks:

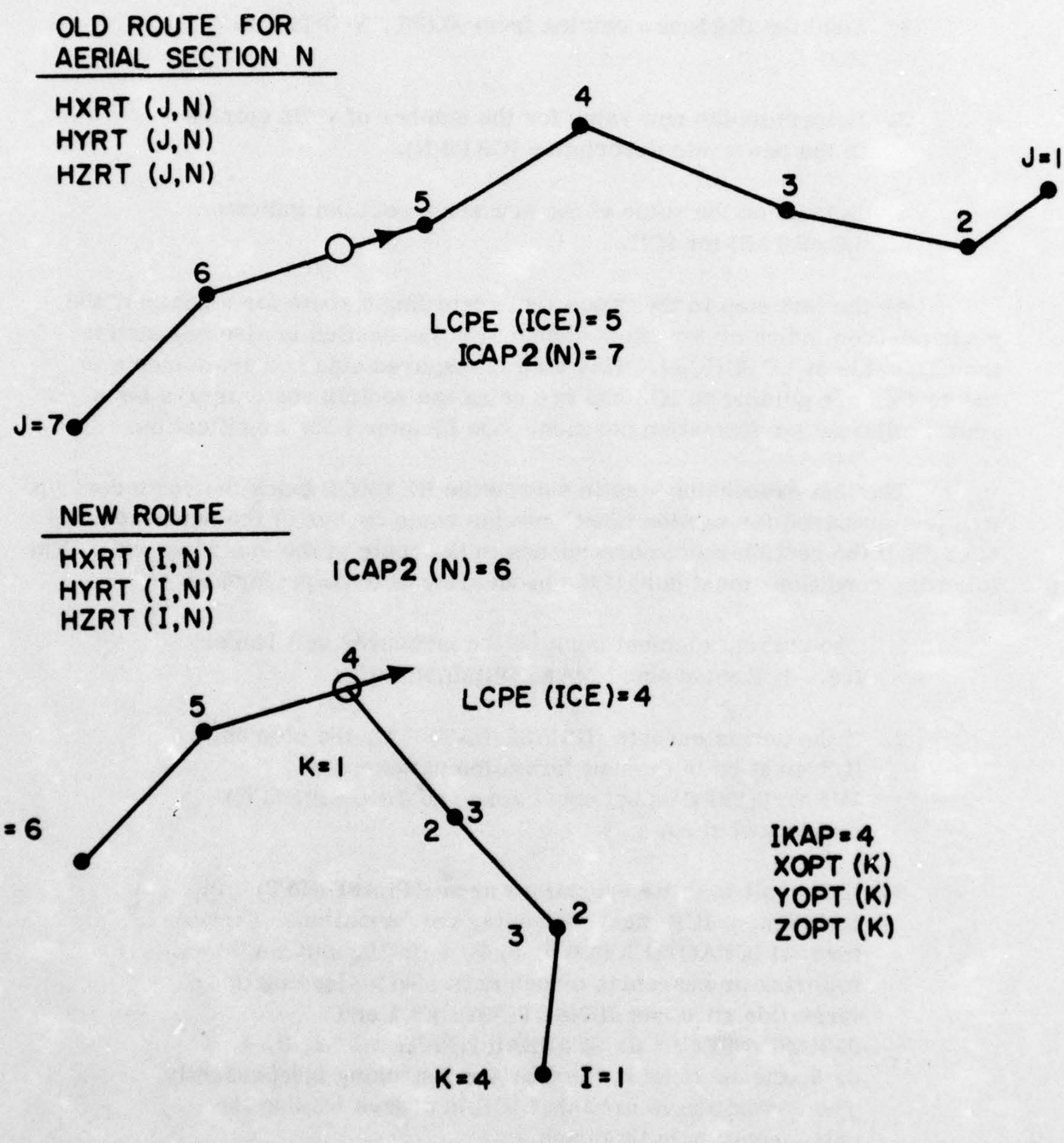


Figure 6.1.--Recording a New Route

1. Determine the number of entries in the old route arrays that have already been traversed. Push these entries "up" or "down" in the arrays so they will be in the proper position after the new entries are recorded.

2. Load the IKAP new entries from XOPT, YOPT, and ZOPT.

3. Determine the new value for the number of valid entries in the new route description ICAP2(N).

4. Determine the value of the new route position indicator LCPE(ICE) for ICE.

As the last step in the process of recording a route for a section, the route position indicator for other elements in the section is also adjusted to the new value of LCPE(ICE). This step is required since other elements in the section are guiding on ICE and are using the section route arrays for a route, adjusted for formation position. See Chapter 7 for clarification.

The last processing step in subroutine HXYMCP loads the route description just prepared for section NSEC into the route arrays of the maneuver unit MANUN if the section route corresponds to the route of the maneuver unit. The following conditions must hold if the processing is to be performed.

1. The current element must be the maneuver unit leader; i.e., ICE must equal MANLDR(MANUN).

2. If the unit is enroute (IPHASE(NAT) = 1), the element ICE must be in the unit formation pattern; i.e., JUNACT(NSEC) must equal zero and JPHASE(NSEC) must equal zero.

3. If the unit is in its operations area (IPHASE(NAT) = 0), the element ICE must be leading the formation. Therefore, if IUNACT(NAT) = 0, 5, 7, 9 or 10, unit NAT is loitering or searching as one unit. To be leading the formation requires JUNACT(NSEC) = 1 and JPHASE(NSEC) = 0. If JUNACT(NAT) = 1, 2, 3, 4, 6 or 8, the sections in the unit are searching independently. The convention is used that ICE is always leading the unit formation in this case.

If ICE is the maneuver unit leader, the conditions above will cause the route of section NSEC to be recorded as the route for MANUN in all cases except when NAT is performing an indirect-fire MISTIC launcher mission and section NSEC is conducting a launch operation.

To load the route arrays for MANUN, the following processing occurs:

1. ICAP2(NSEC) → ICAP1(MANUN).

2. HXRT(I, NSEC) → XRT(I, MANUN)
   HYRT(I, NSEC) → YRT(I, MANUN)   I = 1, . . . , ICAP2(NSEC).
   HZRT(I, NSEC) → ZRT(I, NAT)

   Note that ZRT is subscripted by the aerial unit number NAT since ZRT is an array that is unique to helicopter operations.

Generating a Loiter Station Route

There are two situations that arise in which a route for a section occupying a loiter station is desired. One occurs when a section first arrives at a position over the battlefield where loiter movement is to be conducted. The second occurs when a section that has been loitering reaches the end of its recorded loiter station route. In this case, the route arrays for the section must be reloaded with data.

Both of the above cases are processed in subroutine RTLOIT. The values of IRS are as follows:

$$\text{IRS} = \begin{cases} 3 & \text{an original loiter route is to be constructed,} \\ & \text{and} \\ 4 & \text{loiter route arrays are to be reloaded.} \end{cases}$$

We will discuss the initial route construction process first.

Data defining the characteristics of the loiter station route are contained in six input arrays. Three of the arrays describe a loiter station to be occupied by a section that is operating independently while the other three describe a loiter station to be occupied by the entire unit formation, of which the element selecting the route is the leader.

Both types of stations discussed above are of the shape of a "race track." That is, the station has two "straightaways" of specified length connected by two "turns" of specified radius. The station lies in a plane

of constant altitude and at a specified elevation above the terrain. The orientation and position of the "race track" above the battlefield is initialized so that the current element, at its present position, will lie midway on one of the "turns." Moreover, the initialization process assumes that the section and/or unit will move around the "race track" in a counterclockwise sense.

The input data for a unit loiter station route are as follows:

HALTDU(3, LWC) = desired altitude of the loiter station above the terrain, where

$$LWC = \begin{cases} \text{aerial unit weapon code} \\ \text{LWCOD(NUMELE+NUMART} \\ \quad \text{+MISTUN+NAT) - MWMIS} \end{cases}$$

and the constants used to define LWC are as defined in common area /NUMBER/.

Note that HALTDU is used to define other desired altitudes as will be discussed in other sections. The value of three for the first subscript indicates desired loiter station altitude.

RSTAU(KSUB, LWC) = radius of the loiter station turns where LWC is as defined above and

$$KSUB = \begin{cases} 1 & \text{if a loiter station to be used while waiting for a mission is desired,} \\ 2 & \text{if a loiter station to be used by a unit that has retired is desired,} \\ 3 & \text{if a loiter station to be used by a unit that is occupying a defensive position is desired,} \\ 4 & \text{if a loiter station to be used by a unit conducting an indirect-fire MISTIC launch mission is desired.} \end{cases}$$

XLSTAU(KSUB, LWC) = length of the loiter station sides where LWC and KSUB are as defined above.

Input data for a section loiter station route are arranged in a manner similar to those described above. They are:

HALTDS(3, LCOD) = desired altitude of the loiter station above the terrain where

$$\text{LCOD} = \begin{cases} \text{aerial section weapon code} \\ \text{KWCOD - MAXLWC} \end{cases}$$

and KWCOD is the weapon code of ICE as recorded in /ICECOM/ and MAXLWC is as defined in /NUMBER/.

Note again that HALTDS contains other desired altitudes and that a subscript of three indicates loiter station altitudes.

RSTAS(KSUB, LCOD) = radius of the loiter station turns where LCOD is as defined above and

$$\text{KSUB} = \begin{cases} 1 & \text{if a loiter station to be used by a section while waiting for fire support is desired,} \\ 2 & \text{if a loiter station to be used by a section that has retired is desired, and} \\ 3 & \text{if a loiter station to be used by a section that is occupying a defensive position is desired.} \end{cases}$$

XLSTAS(KSUB, LCOD) = length of the loiter station sides where LCOD and KSUB are as defined above.

Arbitrarily, thirteen new route points are generated by the procedure. Therefore, IKAP = 13.

The first new route point XOPT(1), YOPT(1), ZOPT(1) corresponds to the current element's present position. Then, the next four points correspond to the end points of the loiter station sides, ordered in the way the current element will encounter them. Finally, points six through thirteen correspond to points two through five; for example, XOPT(I) = XOPT(J) where I = J + 4.

Positions of the points two through five are computed according to a procedure that agrees with the loiter station description given earlier.

Details of the processing can be seen in the flow chart of subroutine RTLOIT in Volume 2.

When IRS equals four, route arrays for the section must be reloaded with loiter station route data. The procedure for determining the new route data points to be loaded is quite simple.

Arbitrarily, ten new route points are entered. Therefore, IKAP = 10. Then,

XOPT(I) = HXRT(11 - I, NSEC)
YOPT(I) = HYRT(11 - I, NSEC)
ZOPT(I) = HZRT(11 - I, NSEC)

where I = 1,...,10.

Generating a Cross-Country Route

Situations in which a cross-country route must be selected are as follows:

a. Unit NAT has commenced a new mission.

b. Unit NAT has decided to seek a defensive position.

c. Unit NAT has decided to retire.

d. Unit NAT has decided to seek a position to await a new mission.

e. Section NSEC has decided to seek a defensive position.

f. Section NSEC has decided to retire.

g. Section NSEC has decided to seek a position to await delivery of requested fire support.

h. Section NSEC has decided to rejoin NAT from a defensive position.

In the first four cases, the final destination is the objective of the maneuver unit OBJX(MANUN), OBJY(MANUN). In the last four cases, the final destination is the objective of the section XS(NSEC), YS(NSEC). In all cases, the initial point is the present position of the current element XE, YE as recorded in COMMON/ICECOM/.

The value of IRS that triggers cross-country route selection in subroutine PICKRT is six. When this value is obtained, subroutine RTCROS is called to determine arrays that define the axis of advance for the section or the unit. Then, subroutine RTSELH uses the axis of advance information to determine the route to be followed. We will discuss the concept of axis of advance as used in subroutine RTSELH as we discuss subroutine RTCROS. Then we will describe the method used by subroutine RTSELH to generate a cross-country route.

Subroutine RTCROS

The axis of advance prepared for use by subroutine RTSELH consists of three points and is not to be confused with the axis of advance discussed earlier which consisted of two points. The axis of advance discussed earlier was a recorded axis of advance for the maneuver unit and was prepared primarily for consistency of notation between TAPCOM II and other portions of DYNCOM. The axis of advance used in subroutine RTSELH is recorded only in temporary working arrays and can be for a unit or for a section.

The RTSELH axis of advance is described by the arrays XA, YA, ZA. These arrays contain the X, Y and Z coordinates, respectively, of points that define the orientation of the route selection area that will be discussed in a subsequent paragraph. As indicated, the number of valid entries in the arrays is three and ICAP, contained in common area /ICAP/, is used to store this number. We will discuss the arrays XA and YA first, deferring our discussion of the array ZA until last.

The first point in the axis corresponds to the current element's position as indicated previously. Therefore,

XA(1) = XE, and
YA(1) = YE.

The last point is normally a point at the edge of the unit or section operations area. As indicated previously, this area is described by a circle of specified radius centered at the recorded section or unit objective. The cross-country route terminates at the edge of the operations area because entry into this area is a trigger for the sections within the unit to begin movement activities that are different than cross-country movement.

The radius of the operations area is contained in the input array RADMA which is arranged as follows:

RADMA(I, J) = radius of operations area, where

$$I = \begin{cases} 1 & \text{for an independently operating section loiter station,} \\ 2 & \text{for an attack mission area of operations,} \\ 3 & \text{for an observation mission area of operations, and} \\ 4 & \text{for an indirect-fire MISTIC launcher mission area of operations} \end{cases}$$

$$J = \begin{cases} 1 & \text{for the blue force, and} \\ 2 & \text{for the red force.} \end{cases}$$

Thus, the last point in the RTSELH axis of advance is normally,

$$XA(3) = XO - RMA * \cos(ANG)$$
$$YA(3) = YO - RMA * \sin(ANG)$$

where XO, YO is the objective position (OBJX(MANUN), OBJY(MANUN) or XS(NSEC), YS(NSEC)). RMA is a value selected from RADMA, and ANG is the orientation angle of the axis

$$ANG = \tan^{-1} \left(\frac{YO - YE}{XO - XE}\right) .$$

Note that there are situations in which XA(3), YA(3) are not computed as indicated above. First, the assumption is made that the axis of advance must be at least 100 meters long. Therefore, it may be that XO, YO is too "close" to XE, YE. That is, the distance from XE, YE to the edge of the operations area is less than 100 meters. When this is the situation, no axis is recorded. Instead, the axis arrays are recorded in such a way as to produce an axis of zero length. This will cause the model to recognize that no enroute motion is required. Thus,

$$XA(I) = XE$$
$$YA(I) = YE \qquad I = 1, 2, 3.$$

When the axis is more than 100 meters long, it may be that normal procedures are still not used for computing the axis of advance. The last point may be off the battlefield. That is, one of the relations

XA(3) < 0,
YA(3) < 0,
XA(3) > XMAX, or
YA(3) > YMAX

may hold where XMAX, YMAX are the battlefield dimensions as recorded in common area /MAPCOM/. This situation may arise when a unit mission against an enemy artillery battery is being analyzed. Helicopters are not allowed to leave the battlefield in order to search for a target or to conduct an attack.

Therefore, in this situation, the procedure is to find the intersection of the line that is oriented in the direction of the desired axis with the battlefield boundary. This intersection is then used as XO, YO in the relations for XA(3), YA(3) given previously.

When the axis is found to be more than one hundred meters long, the second axis point XA(2), YA(2) is taken as a point 100 meters distant from XE, YE on the line passing through XA(1), YA(1) and XA(3), YA(3). This procedure is followed regardless of whether XO, YO is taken as the objective point or the intersection of the axis with the battlefield boundary.

Now that the XA, YA arrays have been defined, we may discuss the array ZA. This array defines the elevation of the desired axis, and the entries are associated with the entries in XA, YA.

The values used are taken from the input array HALTDS or the input array HALTDU. Both these arrays were discussed earlier in the description of loiter station elevations. The definitions for the case at hand are:

HALTDS(1, LCOD) = desired elevation of a section axis of advance above the terrain, where LCOD equals the aerial section weapon code as previously defined, and

HALTDU(1, LWC) = desired elevation of a unit axis of advance above the terrain, where LWC equals the aerial unit weapon code as previously defined.

Note that the array HALTDS is used for section NSEC when JUNACT(NSEC)= 4, 5 or 6. The array HALTDU is used in all other cases. Note also that the array entries are elevations above the terrain. The cross-country route

selection process selects a route that is of constant altitude above the terrain. Thus, the procedure for ZA is as follows:

$$ZA(I) = ZR \qquad I = 1, 2, 3$$

where ZR is selected from HALTDS or HALTDU.

Subroutine RTSELH

Subroutine RTSELH uses a dynamic programming algorithm for selecting a cross-country route. This algorithm is essentially the same as that used in subroutine RTSEL which appears in flow chart form in reference 4. Subroutine RTSEL is used to select routes for ground maneuver units and is discussed in detail in reference 5. Here, we will give only a general description of the procedure and point out only those features in subroutine RTSELH that are different from subroutine RTSEL. We will then list the common areas that contain input data used in subroutine RTSELH.

In subroutine RTSEL, the following concepts are important (see Figure 6.2):

1. A control measure, known as the axis of advance, can be specified for a maneuver unit. The axis of advance consists of a sequence of connected line segments from some initial point to the maneuver unit's objective and corresponds to the arrays XA, YA discussed previously for aerial sections.

2. An area, about the axis of advance, known as the route selection area, can be specified. The area has a depth and width which represents the decision maker's planning horizons for route selection purposes.

3. The route selection area can be represented by a matrix of points X_{ij}, Y_{ij} where the point is the j^{th} entry in the i^{th} row of the matrix.

4. Each point, i, j, has nine neighbors; points to which an element at i, j is free to move.

5. With each point i,j is associated a difficulty: e_m, due to known enemy elements m, both within effective firing range and intervisible with i,j; and s_k, due to intervisible enemy strong points k, containing suspected enemy weapons

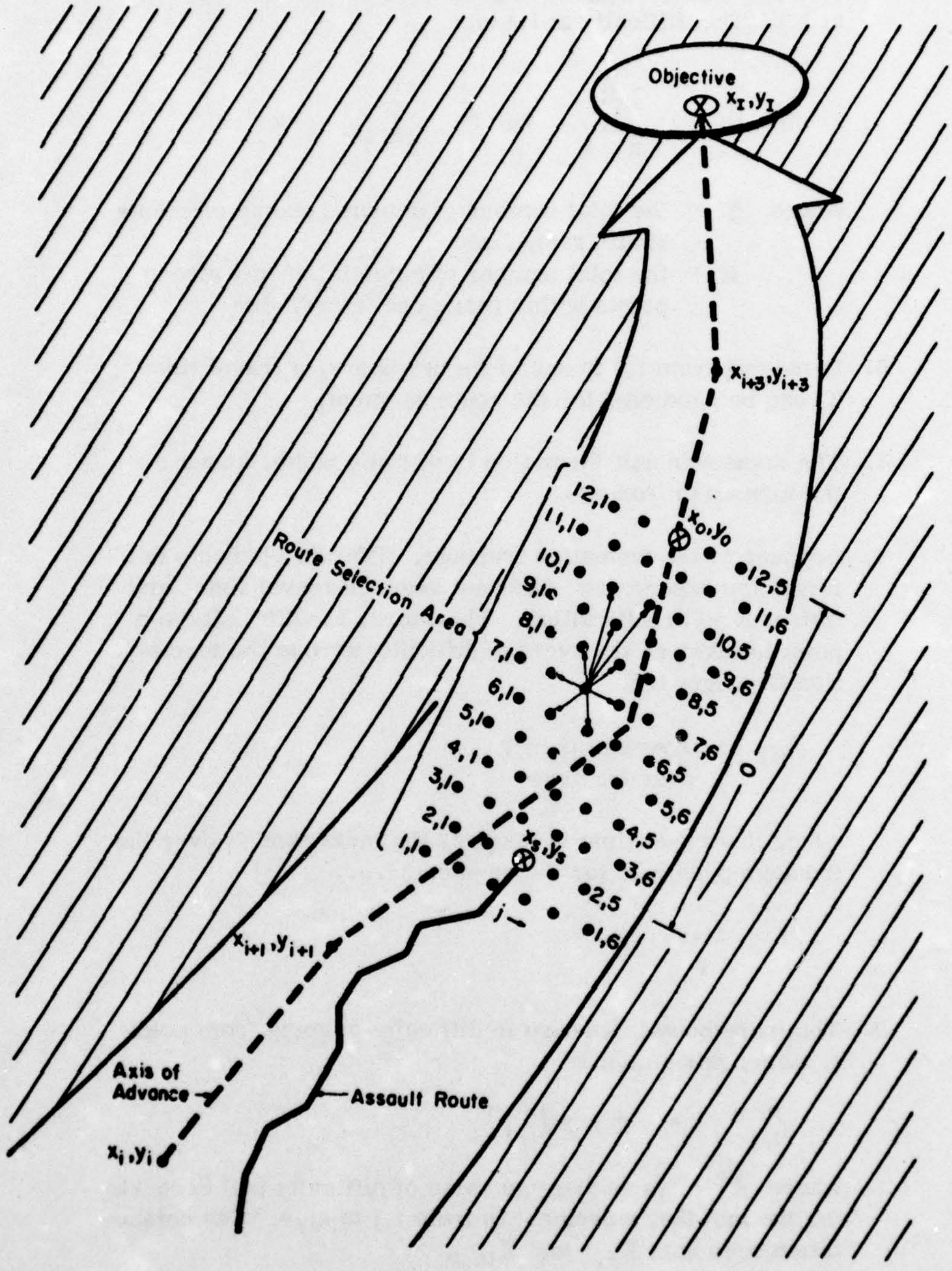


Figure 6.2.--Cross-Country Route Selection Concepts

whose effective ranges are great enough to pose a threat at i,j. The difficulty at i,j is:

$$E(i,j) = \sum_{m=1}^{M} e_m + \sum_{k=1}^{K} s_k$$

where M = the total number of detected enemy elements within range, and

K = the total number of potential enemy strong points within range and intervisible.

6. In moving from i,j to one of its neighbors, a travel time T_f can be predicted for the route segment.

7. The maneuver unit formation is of finite width, known as the formation frontage.

8. Because of the formation frontage, different elements of a formation experience different segment travel times and different point difficulties. Therefore, the difficulty at a point is taken as the average difficulty across the formation frontage; i.e.,

$$\bar{E}_{ij} = \text{Average } (E(ij)) \text{ over frontage}$$

Also, the travel time is taken as the maximum T_f over the frontage predicted for the segment; i.e.,

$$T = \underset{f}{\text{Max}} \ (T_f).$$

9. The incremental increase in difficulty in going from point i,j to neighbor m,n is:

$$d_{i,j}^{m,n} = T + T\bar{\bar{E}}_{i,j}^{m,n}$$

where $\bar{\bar{E}}_{i,j}^{m,n}$ is an average value of difficulty that accounts for the fact that movement is from i,j to m,n. The computation uses both $\bar{E}_{i,j}$ and $\bar{E}_{m,n}$.

This formulation of the incremental difficulty of a route segment implies that a compound effect between travel

time and exposure to enemy weapons exists. Minimum difficulty is achieved when both travel time and the time of exposure to a given set of enemy weapons is reduced.

In general, the route selection area matrix is constructed so that the leader is initially at the third point of row 2. A line of length ϕ is constructed from the leader to a point on the axis of advance and this point becomes the third entry of row 12. Spacing between rows, and between points of a given row, is $\phi/10$ or EMIN. However, if the objective is closer to the leader than distance ϕ, rows are deleted from the matrix. This procedure is followed because it is desired to have the objective appear as the third entry of the last row and because it is desired to maintain a distance between rows of approximately EMIN. Therefore, rows are deleted two at a time until the adjusted distance between rows is as close to EMIN as possible.

The dynamic programming algorithm is based upon the recursive relation

$$_{m,n}d_{i,j} = {_{a,b}d^*_{m,n}} + d^{i,j}_{m,n}$$

where

$_{a,b}d^*_{m,n}$ = lowest tactical difficulty to reach (m,n) from the starting point of the route given that the route goes through the point (a,b) immediately prior to reaching (m,n).

$_{m,n}d_{i,j}$ = total tactical difficulty to reach (i,j) from the starting point of the route given that the route passes through the point (m,n) immediately prior to reaching (i,j).

For computational purposes, the program maintains two lists. In list U, the grid points (m,n), to which the minimum difficulty path has been determined, are recorded. Associated with each grid point (m,n) in list U, the value of the minimum difficulty $_{a,b}d^*_{m,n}$ and the entry point (a,b) are also recorded. In list V are recorded the nine neighbor points (i,j) of each point (m,n) in list U. However, list U and list V must be disjoint, so each neighbor point of (m,n) already appearing in list U is omitted from list V. Now, for each grid point (i,j) contained in list V, the value of the total tactical difficulty $_{m,n}d_{i,j}$ and the entry point (m,n) are also recorded. Note that (m,n) always appears in list U.

The use of these two lists becomes apparent when the computational procedure of the dynamic programming algorithm is outlined.

1. Enter the start point (2, 3) in list U and record the minimum difficulty to reach (2, 3) from (2, 3) as zero; i.e., set ${}_{2,3}d^*_{2,3} = 0$ and record (2, 3) as an entry point.

2. Compute the difficulties $d^{i,j}_{2,3}$ to reach each of the nine neighbors (i, j) of (2, 3) and record the difficulty of points (i, j) in list V. Also, record the entry point (2, 3).

3. Search all values $d^{i,j}_{2,3}$ associated with the entries (i, j) in list V for the minimum entry i, j; that is, find (s, t) such that

$$ {}_{2,3}d^*_{s,t} = \underset{(i,j) \in V}{\text{Min}} \left(d^{i,j}_{2,3} \right) . $$

   Enter (s, t) in list U and record ${}_{2,3}d^*_{s,t}$ and the entry point (2, 3). Remove (s, t) from list V to maintain disjoint sets.

4. Record the difficulty of points (i, j); i.e., the difficulty of the neighbors of (s, t) in list V. Compute ${}_{s,t}d_{i,j}$ associated with each neighbor (i, j) and record (s, t) as the entry point. Delete this step for any (i, j) already appearing in list U.

5. Search the entire list V for the entry having the minimum recorded total tactical difficulty. Denoting this point by (s, t), enter (s, t) in list U and remove from list V. Also, record the minimum tactical difficulty ${}_{m,n}d^*_{s,t}$ where (m, n) is the associated entry point. Repeat step 4.

The first point in the last row of the route selection area matrix that appears in list U defines the least difficult path to a point in the last row. So long as the unit's objective does not fall within the route selection area, this path is accepted as the desired route, and the coordinates of the route points are computed from the grid points contained in list U. In the event the unit's objective falls within the route selection area, the computational procedure outlined above is repeated until the center point of the last row of the route selection area matrix appears in list U. This procedure ensures that the unit's route does, in fact, go to the objective.

There are several important differences in procedure between subroutine RTSEL and subroutine RTSELH. First, the axis of advance as represented by XA, YA, always consists of a straight line. The leader's position when a decision is made to perform enroute movement is the initial point in the axis of

advance XA(1), YA(1). The objective point of the section is the final point XA(3), YA(3). The remaining point XA(2), YA(2) is on the line from XA(1), YA(1) to XA(3), YA(3).

Next, the route selection area always extends from one end of the axis of advance to the other and the matrix normally includes twenty (20) rows. If the route selection area of depth ϕ includes the objective, rows are deleted from the selection area matrix as is done in subroutine RTSEL. However, if the route selection area of depth ϕ does not include the objective, the area is stretched both in depth and width. Twenty rows are maintained in the matrix, but the distance between rows EMIN increases. Entries of a given row also become more widely separated. These entries are also EMIN apart. The implications of these procedures are:

1. The leader always plans the section's route all the way to the objective;

2. A greater planning horizon in depth is accompanied by a greater planning horizon in width which allows the section more lateral maneuverability on longer routes; and

3. Resolution of the route-selection process is bounded from above and may decrease significantly for longer routes.

This last implication is unfortunate but difficult to avoid from the standpoint of storage requirements and computation time. However, the reduced resolution is probably representative of actual practice.

The next difference in procedure is associated with the concept of formation frontage. In subroutine RTSELH, the assumption is made that the formation is of zero width. Thus, the difficulty at point (i, j) due to known enemy weapons and suspected strongpoints is computed simply as

$$\overline{E}_{ij} = E(i,j) = \sum_{m=1}^{M} e_m + \sum_{k=1}^{K} s_k .$$

Similarly, the travel time T is simply $T = T_f$. Now, the method used to compute T_f is also different. In RTSEL, the mechanical characteristics of the vehicle, the mobility characteristics of the soil, and the gross terrain profile, are used to estimate the time required to travel a particular segment. In RTSELH, the desired enroute speed of the section, SPDSE(NSEC), is used as a constant for all route segments. Thus, the estimated movement time is simply proportional to the segment length. This difference in procedure is acceptable

since aircraft in flight tend to move at much nearer a constant speed than do ground vehicles negotiating terrain of widely varying characteristics.

Next, the method used to compute $\bar{\bar{E}}_{i,j}^{m,n}$ in the expression

$$d_{i,j}^{m,n} = T + T\bar{\bar{E}}_{i,j}^{m,n}$$

is also different in subroutine RTSELH. The method accounts for the fact that in movement from one point to another point (not necessarily in the next row), possibly as many as two intervening rows are crossed. In moving through these rows, no points in the rows are actually encountered. Instead, movement is between two adjacent difficulty points in the rows. Thus, $\bar{\bar{E}}_{i,j}^{m,n}$ is taken as the average of difficulties encountered along a segment and these difficulties may include intervening "row difficulties." For example, in Figure 6.2:

$$\bar{\bar{E}}_{7,3}^{8,2} = 1/2\ (\bar{E}_{7,3} + \bar{E}_{8,2}),$$

$$\bar{\bar{E}}_{7,3}^{9,3} = 1/3\ (\bar{E}_{7,3} + 1/2\ (\bar{E}_{8,2} + \bar{E}_{8,3}) + \bar{E}_{9,3}),$$

and

$$\bar{\bar{E}}_{7,3}^{10,2} = 1/4\ (\bar{E}_{7,3} + 1/3\ (2\bar{E}_{8,2} + \bar{E}_{8,3})$$

$$+ 1/3\ (\bar{E}_{9,2} + 2\bar{E}_{9,3}) + \bar{E}_{10,2}).$$

Next, in subroutine RTSELH, the route starts from the matrix point ISTRT, JSTRT and proceeds to the matrix point IFIN, JFIN. Thus, the start point does not have to be the point (2, 3), and the final point does not have to be in the last row. However, several contraints are placed on the selection of values for the four variables defined above. The constraints are:

1. ISTRT must be even;

2. IFIN must be even; and

3. JFIN must equal JSTRT.

The flexibility introduced by the changes cited above permit the user a wider latitude in his representation of the route selection process. However, the user should note that if IFIN is selected to be less than twenty (20), the route selected may not extend all the way to the objective. This situation will occur any time the number of rows determined for the difficulty matrix exceeds the value entered for IFIN.

Finally, the determination of E(i, j) is different for aerial vehicles. A ground vehicle at a particular point on the battlefield has the elevation that is unique to that terrain position. Thus, the known enemy elements and suspected strongpoints considered at point (i, j) are those that are intervisible from the vantage point afforded by the terrain elevation at (i, j). For aerial vehicles, the altitude of the vehicle must be taken into account. Thus, those elements and strongpoints considered at point (i, j) are those that are intervisible from the vantage point afforded by the vehicle's elevation. The reader will recall that aerial vehicles fly at constant altitude above the terrain profile; that is, they fly "nap of the earth." Therefore, vehicle elevation is determined from terrain elevation by adding a constant. The constant is any entry in the axis of advance array ZA.

Subroutine RTSELH appears in flow chart form in Volume 2. For further explanation of the concepts used in route selection and the procedures used in subroutine RTSEL, see reference 5. The reader should note that the battlefield coordinates of a point in the difficulty matrix are computed in subroutine XYLOCH. Moreover, the computation of $d_{i,j}^{m,n}$ and hence, T and $\overline{E}_{i,j}^{m,n}$ is performed in subroutine HDIF.

The variables that must be input for use by the subroutines cited above are contained in the common areas listed below. The variables contained in these common areas are defined in the descriptions that appear in Volume 2. or in reference 3.

COMMON/ESMH/
COMMON/ET/
COMMON/EW/
COMMON/LWCOD/
COMMON/RTKONH/
COMMON/RTSIZE/
COMMON/S/
COMMON/SCAPH/
COMMON/SPTS/
COMMON/T/
COMMON/TGTDIM/

Generating a Search Route

A search route for a section is required any time a section enters a mission operations area and the unit is conducting an attack or observation mission. Thereafter, a new search route must be selected each time the section concludes an attack or reaches the end of its previously recorded search route. In subroutine PICKRT, two is the value of IRS that indicates that a search route is desired, and subroutine RTSRCH is used to generate the route description.

There presently exists no experimental or operational evidence upon which to base the procedure used to select a search route in subroutine RTSRCH.

As currently formulated, the model is based upon hypotheses derived from current doctrine and discussions with helicopter pilots having operational experience. The model only approximately represents search route selection decision processes. Carefully planned experiments should be conducted to identify the underlying decision process used during search and to verify the decision concepts involved.

The hypotheses of the search route generation model are as follows:

1. Search is conducted for targets in an area that has definite and finite boundaries. This hypothesis is based upon the fact that an aerial unit nearly always conducts a mission in response to a fire request from an element that has seen or suspects enemy targets. In preparing the fire request, the requesting element implies a limited area to be searched for the targets in question, and the location of this area is part of the fire-request message.

2. In searching the area defined by hypothesis one, the elements of an aerial section tend to sequentially search points of the terrain that are likely positions for enemy targets to be located. These points are selected by decision processes that consider such factors as vegetation and terrain characteristics in the target area, the reported target coordinates and other factors related to training and doctrine.

To implement the results of hypothesis one above, the search area is viewed in the model to be a circle of specified radius centered at the reported target location. Thus, the center of the area in question is at OBJX(MANUN), OBJY(MANUN). The radius of the area is contained in the array RADMA discussed previously. That is,

RADMA(I,KP1) = radius of the search area, where

$$I = \begin{cases} 2 & \text{if the unit is conducting an attack mission,} \\ 3 & \text{if the unit is conducting an observation mission, and} \end{cases}$$

$$KP1 = \begin{cases} 1 & \text{for the blue force, and} \\ 2 & \text{for the red force.} \end{cases}$$

In the absence of experimental evidence about the decision processes used to select sequential search points, hypothesis two is more difficult to implement. As one solution, we have adopted the following procedure.

1. Select a point within the target search area whose coordinates are distributed according to a conical density function.

2. If the point is closer than desired to a previously selected search point, discard the point.

3. Continue selecting points by the method of steps 1 and 2 until an adequate number have been selected.

The above procedure at least recognizes that search tends to be concentrated about the reported target location. It also allows for the fact that sequential search positions should be somewhat separated to reduce coverage overlap between search positions and to prevent abrupt changes in heading during search. The reader should note, however, that methods have not yet been developed to account for target location clues provided by vegetation or terrain or for special search techniques used as a result of training or doctrine. Thus, the method of randomly selecting search positions is used.

Actually, there is also another criterion that must be met by a search route point that is not specified by the procedure outlined above. Each point must be over the battlefield since helicopters are not allowed to leave the battlefield in TAPCOM II.

The procedure used in subroutine RTSRCH also assumes that the search route is at constant elevation above the terrain. Thus, as the X and Y coordinates of a search route point are selected, the elevation of the terrain at the point must be determined and added to the constant search elevation to obtain the altitude above the zero elevation plane of the search route elevation at the point in question. That is, for point I at XOPT(I), YOPT(I)

ZOPT(I) = HALT + ELVATE(XOPT(I), YOPT(I), 0)

where function ELVATE returns the terrain elevation and HALT is the specified search elevation. The variable HALT is obtained from the arrays HALTDS or HALTDU which have been discussed before. We have,

HALT = HALTDS(2, LCOD) for an aerial section with weapon code LCOD searching independently for targets, or

HALT = HALTDU(2, LWC) for an aerial unit with aerial unit weapon code LWC searching for targets (IUNACT(NAT) = 7).

The procedure for selecting XOPT(I), YOPT(I) from the conical density function is as follows:

1. Set XO = OBJX(MANUN), and set YO = OBJY(MANUN).
2. Select two random numbers R1 and R2 from a uniform density on the interval (0, 1).
3. Compute a random angle for the I^{th} route point, relative to XO, YO; i.e., ANG = $2 \cdot \pi \cdot$ R1.
4. If R2 = 0, set X = 0 and go to step 10.
5. If R2 = 1, set X = 1 and go to step 10.
6. Compute $\cos(\phi) = 1 - 2 \cdot R2$.
7. Compute $\sin(\phi) = \sqrt{1 - \cos^2(\phi)}$.
8. Compute

$$\theta = 1/3 \tan^{-1}\left(\frac{\sin(\phi)}{\cos(\phi)}\right) + 4/3\pi .$$

9. Compute $X = \cos(\theta) + 1/2$.
10. Compute R = X * RMA where RMA is as selected from RADMA.
11. Compute

$$XOPT(I) = XO + R\cos(ANG)$$
$$YOPT(I) = YO + R\sin(ANG).$$

12. The procedure is complete.

The above procedure provides a point representing a random sample from a conical distribution whose density function is

$$dF(R, ANG) = \frac{3R}{\pi * RMA3} (RMA - R)\, dR\, dANG.$$

The conditional distribution of R, given ANG, is identical to the marginal distribution of R and is specified by the relation

$$f(R) = \frac{6R}{RMA^3} (RMA - R).$$

The marginal distribution of ANG is uniform and is specified by the relation $g(ANG) = 1/2\pi$. The procedure outlined above is actually a Monte Carlo sampling scheme which utilizes the inverses of the distribution functions corresponding to f(R) and g(ANG). The inverse of the distribution function corresponding to f(R) is obtained from reference 6, page 393.

In subroutine RTSRCH, ten points are selected for the route description. Therefore, IKAP = 10. The points that are selected are those that satisfy the criteria cited earlier. That is, a point I must be on the battlefield implying $0 \le XOPT(I) \le XMAX$ and $0 \le YOPT(I) \le YMAX$ where XMAX and YMAX are the battlefield dimensions recorded in common area /MAPCOM/. Furthermore, the point I cannot be "too close" to point I - 1. In the model, this implies that the relation

$$\sqrt{(XOPT(I)-XOPT(I-1))^2 + (YOPT(I)-YOPT(I-1))^2} \ge 100$$

must hold for point I to be included.

<u>Generating an Attack Route</u>

From decision analyses discussed in previous sections of this chapter, an attack route for an aerial section is selected in order that the section may conduct a direct or indirect-fire attack or illuminate targets within a target complex for indirect-fire MISTIC missiles launched by some other element. Subroutine RTATAK is designed to select such routes.

In general, an attack route consists of three segments as shown in Figure 6.3. The first segment leads from the position of the section leader at the time the attack route is selected (point 1) to a point over the battlefield known as the initial point or IP (point 4). The second segment is known as attack phase one and extends to a point over the battlefield known as the end point or EP (point 6). Point 4 is at range RIP from the target while point 6 is at range REP. The third segment is known as attack phase two and extends from point 6 to the target (point 8). The points 2, 3, 5, and 7 will be discussed in a subsequent paragraph.

As discussed previously, if a direct-fire attack is being conducted, elements within the section may be firing a variety of suppressive and point-fire

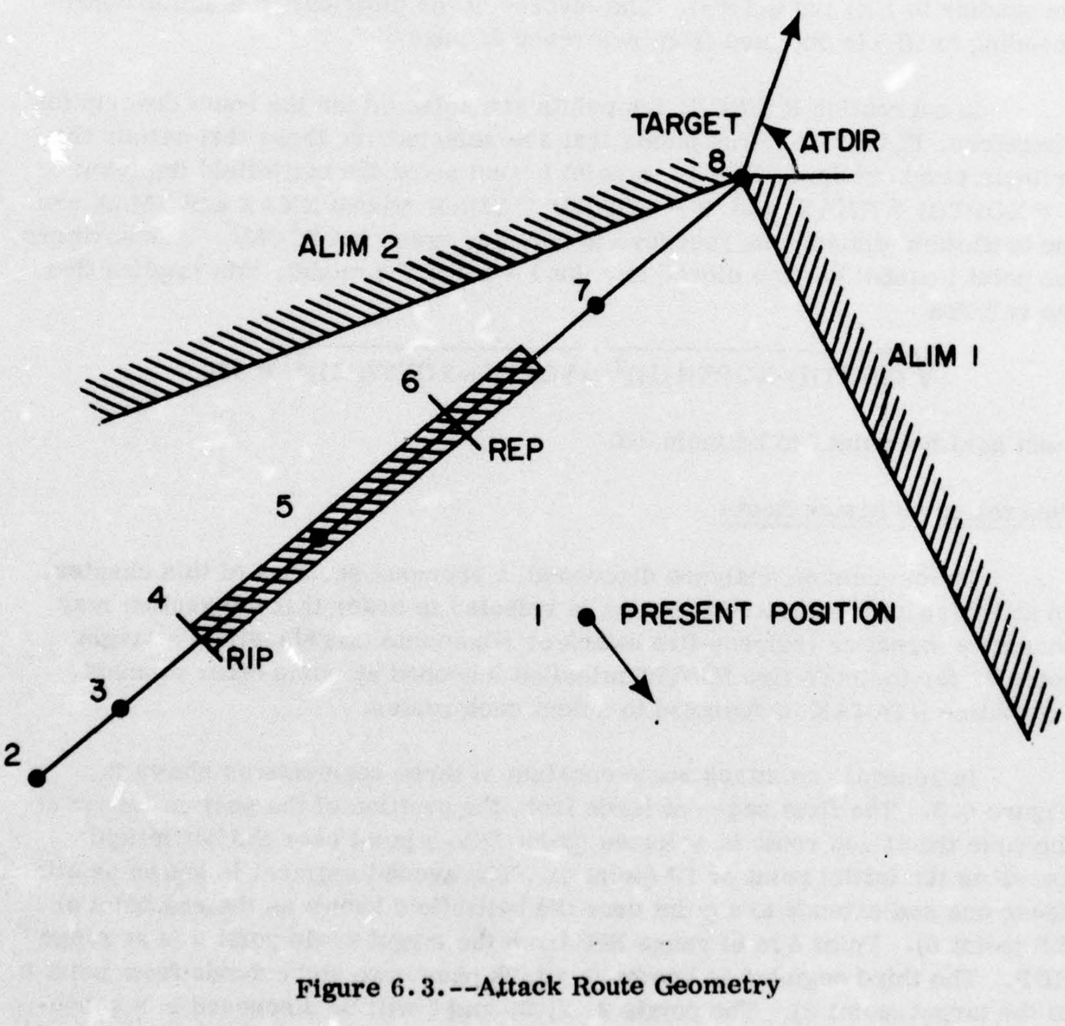


Figure 6.3.--Attack Route Geometry

weapons during attack phase one. The strategy selected for these firing activities specifies the length of this attack segment. During attack phase two, elements within the section fire only suppressive fire weapons while point-fire weapons launched during attack phase one are still flying. The segment terminates when all point-fire weapons have impacted. Thus, phase two will never actually extend to the target. Both attack phases are illustrated by the shaded line in Figure 6.3.

During an indirect-fire MISTIC attack, a specified number of MISTIC missiles are launched, with a specified time interval separating each launch. The first missile is launched as soon as the IP is reached and the launches continue until the attack is complete. As much of the attack route as is needed between points 4 and 8 is used during the attack.

When a section is illuminating targets within a target complex, illumination is assumed to commence at the IP and to continue for so long as indirect-fire MISTIC missiles previously requested for launch from some other element are still flying. Thus, as in the case of the MISTIC launcher attack, as much of the attack route as is needed between points 4 and 8 is used.

From the above general discussion, we see that the ranges RIP and REP are critical to the determination of the details of the attack route. The methods used to compute these ranges are different depending upon the type of route desired.

For illumination routes, the relations RIP = RFOMAX(LWC) and REP = 0 are used, where LWC is the MISTIC weapon code of the MISTIC unit to which section NSEC belongs. The array RFOMAX is input and specifies the range at which initiation of illumination is desired for each MISTIC unit weapon code. From these values for RIP and REP, we see that the phase one route is of length RIP and that the length of the phase two route is zero.

For indirect-fire MISTIC attacks, we use the relations RIP = RLNMAX(LWC) and REP = 0 where LWC is again the MISTIC unit weapon code. The array RLNMAX is input and specifies the range at which an indirect-fire MISTIC launch sequence should commence for each weapon code. Again, the length of phase one is RIP and the length of phase two is zero.

For direct-fire attacks, the computation of RIP and REP is more complex. In general, REP is computed first and then RIP is found by adding a range increment R to account for the speed of the section during the attack and the duration of attack phase one.

The range REP is computed as follows:

$$REP = \max_{J \in ISEC} (RFMIN(J)).$$

Here, RFMIN(J) is the minimum range at which element J can fire a weapon during attack phase one, and the maximization process for REP above assumes that no element in section ISEC will violate its minimum firing range constraint.

The variable RFMIN(J) is taken from the input array RDFMIN. This array is defined as follows:

RDFMIN(I, LCOD) = minimum range at which weapons may be fired during attack phase one, where LCOD equals the aerial section weapon code of section ISEC, and

$$I = \begin{cases} 1 & \text{if element J is to fire a direct-fire MISTIC missile during phase one,} \\ 2 & \text{if element J is to fire a beam-rider missile during phase one,} \\ 3 & \text{if element J is to fire any other type of ballistic point-fire weapon during phase one, and} \\ 4 & \text{if element J is to fire only suppressive fire weapons during phase one.} \end{cases}$$

To determine the proper value of I above, we use the arrays IHAMO and IHTARG which are prepared by the target assignment model discussed previously. We also use the input array IHDFMC which is defined as follows:

IHDFMC(K, LCOD) = ammunition code of weapon type K aboard a helicopter with aerial weapon code LCOD, where

$$K = \begin{cases} 1 & \text{for MISTIC missile,} \\ 2 & \text{for beam-rider missile, and} \\ 3 & \text{for any other type of ballistic point-fire weapon.} \end{cases}$$

The reader may view the details of the procedure for computing the subscript I in RDFMIN(I, LCOD) by consulting the flow chart of subroutine RTATAK in Volume 2.

The range increment R, used with REP to determine RIP, is computed by the relation R = TDUR * SPDSE(NSEC) where SPDSE(NSEC) is the desired speed of section NSEC as discussed previously. The variable TDUR is the duration of attack phase one and is computed by the relation

$$\mathrm{TDUR} = \max_{J \in \mathrm{ISEC}} (\mathrm{TDMIN(J)}).$$

Here, TDMIN(J) is the duration of firing desired by element J during attack phase one, and the maximization process assures that all elements in section ISEC will be allowed sufficient time for firing.

The duration TDMIN(J) for element J is computed by one of three methods which are outlined below.

If the element is to fire only a point-fire weapon during phase one, then the relation TDMIN(J) = TFLY(L) is used, where L is the helicopter number of element J and TFLY(L) is the time required to fire the point-fire weapon as discussed in a previous section.

If the element is to fire a suppressive fire weapon as well as a point-fire weapon during phase one, then the relation

$$\mathrm{TDMIN(J)} = \mathrm{FACTL(5, L)} + \mathrm{FACTL(4, L)} + \mathrm{TFLY(L)}$$

is used. The array FACTL is prepared by subroutine WASAIR. The first entry used in the above relation gives the total amount of time required to fire all bursts from the suppressive fire weapon. The second entry gives the length of a pause in firing that occurs between the suppressive fire bursts and the preparation of the point-fire weapon.

Finally, if suppressive fire weapons are the only weapons to be fired by element J during phase one, then TDMIN(J) is initially assigned a value of zero. If, after the maximization process for R, it is found that all elements in the section are to fire only suppressive fire weapons (R = 0), then the relation TDMIN(J) = 2 * EVHTIM(LCOD) is used for all elements in section ISEC. In this case, TDUR will also be 2*EVHTIM(LCOD). The array EVHTIM is input and gives the length of a standard event for a helicopter section with weapon code LCOD. The above procedure simply states that the duration of attack phase one is two standard events when all helicopters in the section are firing only suppressive fire weapons.

The points 2, 3, 5, and 7 shown in Figure 6.3 are included only to permit smooth operation of the movement model reported in Chapter 7. They help

define the shape of the route in finer detail. The points are:

1 at range RIP + 300 (point 2)
2 at range RIP + 150 (point 3)
3 at range (RIP + REP)/2 (point 5)
4 at range REP/2 (point 7).

Orientation of the attack route must be specified, both in azimuth and in zenith. We will discuss the azimuth of the attack route first.

For illumination, indirect-fire MISTIC, and the first direct-fire attack conducted against a specific target complex, the line containing points 2 through 8 passes through point 1. The rationale behind this procedure is that the most important consideration in the situations cited above is the time required to bring the target under fire. A procedure is desired that will tend to produce minimum reaction times and maximize the "surprise factor." While the true minimum reaction time would be achieved by a procedure which considers the direction of heading and turn capability of the attacking section, the stated procedure is adopted as an adequate approximation.

In the case of a subsequent direct-fire attack on a specific target complex, the direction of the attack is selected to be within the attack sector defined by the shaded limits ALIM1 and ALIM2 shown in Figure 6.3. The included angle between these limits is input and the sector is oriented so that all subsequent attacks originate from "the friendly side."

The input data required to define the attack sector discussed above are as follows:

ATLIM(I) = included angle of the allowable attack sector, where

$$I = \begin{cases} 1 \text{ for the blue force, and} \\ 2 \text{ for the red force.} \end{cases}$$

ATDIR(I) = orientation angle for the allowable attack sector, where

$$I = \begin{cases} 1 \text{ for the blue force, and} \\ 2 \text{ for the red force.} \end{cases}$$

The latter array entries correspond to the angle that a vector bisecting the attack sector would make with the battlefield X axis (see Figure 6.3). The

vector is oriented to point in the direction of allowable attacks, and the angle is defined on the interval $(-\pi, \pi)$.

The actual attack direction is chosen by a Monte Carlo sample from a uniform distribution over the sector. The rationale for this procedure is summarized by the following assumptions.

1. The objective of subsequent direct-fire attacks on a target is to produce satisfactory destruction while maximizing the chances of attacker survival.

2. Any attack route chosen by the stated procedure will yield satisfactory destruction.

3. Maximum survival chances are achieved by maximizing the "surprise factor" of the attack while avoiding flight over known enemy territory.

4. Randomly distributed attack routes from "the friendly side" are adequate approximations that achieve the objective of assumption 3.

The zenith angle of an attack route is different depending upon the type of attack being conducted. The model formulates a route for indirect-fire MISTIC attacks that is of constant altitude above the zero elevation plane. Thus, the zenith angle BETA is equal to zero. This type of route is assumed to produce the most effective and stable attitude for the launch platform during the MISTIC indirect-fire launch sequence.

For illumination and direct-fire attacks, however, the attack route is constructed so that it passes through a specified altitude above the target at the IP. Therefore, BETA for these types of attacks is nonzero.

From the discussion above, points 2 through 8 (Figure 6.3) for an indirect-fire MISTIC attack route are all at the same elevation above the zero level. This altitude is ZO and is found from the relation

$$ZO = ELVATE(XO, YO, 0) + HALT.$$

Function ELVATE returns the elevation of the terrain at the target position XO, YO. The variable HALT is the desired elevation of the attack route above the target. It is found by the relation HALT = HALTDS(5, LCOD) where LCOD is the aerial section weapon code and the array HALTDS has been discussed previously.

The target position XO, YO is recorded in the arrays XD, YD as follows:

XO = XD(NF)
YO = YD(NF)

where NF is the fire-support firer number corresponding to the aerial section's MISTIC launcher number. The arrays XD, YD as used above are discussed in Chapter 3.

For illumination and direct-fire attacks, the attack route passes through the elevations ZO and ZIP. ZO is the elevation of the target above the zero level while ZIP is at range RIP from the target (Figure 6.3) and is defined by the relation ZIP = ZO + HDES where HDES = HALTDS(KSUB, LCOD). Here, the subscript LCOD is again the aerial section weapon code, and the subscript KSUB is defined as

$$\text{KSUB} = \begin{cases} 4 & \text{if a direct-fire attack is being conducted,} \\ & \text{and} \\ 6 & \text{if an illumination route is being constructed.} \end{cases}$$

Since ZIP is at range RIP from the target, the zenith angle BETA is found by the relation

$$\text{BETA} = \sin^{-1}\left(\frac{\text{HDES}}{\text{RIP}}\right).$$

The elevation ZO is found by function ELVATE with the target position coordinates XO, YO being used as calling parameters. For illumination, the coordinates are defined by the relations

XO = XDFO(NFO)
YO = YDFO(NFO)

where NFO is the forward observer number assigned to the aerial section. The arrays XDFO, YDFO are discussed in reference 1.

For direct-fire attack routes, the coordinates XO, YO are determined as the centroid of locations of elements assigned as targets to the members of aerial section NSEC. From previous discussions, it is possible for a single element I to have as many as two assigned targets. The first, if any, target is MDFAF(I) while the second is LTARG(I). However, the second target is not included in the centroid computation if it is the same target as the first; i.e., if LTARG(I) = MDFAF(I). Also, there will be no targets assigned to element I if MDFAF(I) = 0.

Now, the reader should note that all eight points defined in Figure 6.3 are determined at the time that a decision to select an attack route is made. The points are stored in the arrays XSAVE, YSAVE, ZSAVE for later use. The arrangement for each array is identical and is illustrated by that for XSAVE; i.e.,

XSAVE(I, J)

where

I = route point number ($1 \leq I \leq 8$), and

J = aerial section number.

The procedure for computing the array entries for section NSEC is presented in the flow chart of subroutine RTATAK that appears in Volume 2. The details will not be repeated here but they are based on the concepts that been discussed previously.

Subroutine RTATAK is called in subroutine PICKRT when IRS is equal to 5, 7, 8 or 10. When IRS is equal to 5 or 10, the arrays XSAVE, YSAVE and ZSAVE are filled by either the procedure for an initial attack or that for a subsequent attack. Then, the route arrays for section NSEC are filled with the information that describes the route to the initial attack position. That is,

XOPT(I) = XSAVE(I, NSEC)
YOPT(I) = YSAVE(I, NSEC)
ZOPT(I) = ZSAVE(I, NSEC)

where I = 1, 2, 3, 4.

When IRS is equal to seven, the route for attack phase one is desired so the only processing required is

XOPT(J) = XSAVE(I, NSEC)
YOPT(J) = YSAVE(I, NSEC)
ZOPT(J) = ZSAVE(I, NSEC)

where J = I - 3 and I = 4, 5, 6.

Finally, when IRS is equal to eight, the only processing required is:

XOPT(J) = XSAVE(I, NSEC)
YOPT(J) = YSAVE(I, NSEC)
ZOPT(J) = ZSAVE(I, NSEC)

where J = I - 5 and I = 6, 7, 8.

In the first case above, the number of valid entries in the route arrays is four (IKAP = 4), while in the other cases, the number of valid entries is three (IKAP = 3).

Generating a Transition Route

A transition route for an aerial section is one which is used by a section that is attempting to achieve its proper position in a unit formation. The unit can be searching for targets (IUNACT(NAT) = 7 and IPHASE(NAT) = 0), loitering (IUNACT(NAT) = 0, 5 or 9 and IPHASE(NAT) = 0) or performing enroute movement (IPHASE(NAT) = 1). The section can be returning to the formation from an indirect-fire attack or a defensive position, the section can be joining the formation of a unit that has just arrived at the section's defensive position or the section can be joining the unit formation as it commences enroute movement. In all cases, the section has a position which it desires to achieve in the formation and which is specified by input unit organization data. In general, the end point of the transition route will be constantly in motion since the formation is in motion. Therefore, a section that is moving along a transition route will choose a new route each event of the section leader.

The value of one for IRS triggers transition route selection in subroutine PICKRT. Subroutine RTJOIN is then used to generate the transition route. We will discuss the concepts employed in this subroutine in the paragraphs which follow.

The first operation in subroutine RTJOIN is concerned with determining the element that is leading the formation. This information is required so that the leader of the section being analyzed may select its formation position relative to this lead element. The processing is accomplished in subroutine FRMLDR.

Normally, the unit formation leader is the element designated as the maneuver unit leader; i.e., LDR = MANLDR(MANUN). However, it may be that the section containing the maneuver unit leader may be conducting an indirect-fire MISTIC attack at the time that a formation leader is determined. In this case, some other element will be leading the unit formation.

The element that is leading the formation is the one that appears highest in the unit leadership hierarchy described earlier in our discussion of subroutine NATLDR and that is also flying in the unit formation. The latter criterion means that JPHASE(NSE) = 0 and either JUNACT(NSE) = 1 if IPHASE(NAT) = 0 or JUNACT(NSE) = 0 if IPHASE(NAT) = 1 where NSE is the aerial section number of the leadership candidate.

The leadership hierarchy is as follows:

1 Leader of section one in platoon one
2 Leader of section two in platoon one
3 Leader of section one in platoon two
4 Leader of section two in platoon two
:
2N-1 Leader of section one in platoon N
2N Leader of section two in platoon N.

Of course, it is possible that the maneuver unit organization has been specified initially with missing sections and platoons. However, it is also possible that as many as fourteen elements would exist in the hierarchy because there are as many as two sections in a platoon and as many as seven platoons in the unit ($N \leq 7$). In any event, subroutine FRMLDR returns the proper lead element as LDR, and if no element in the leadership hierarchy is flying with the unit, then LDR is returned as ICE. This convention is used to indicate that the current element must act as its own leader. This case occurs if and only if ICE is attempting to enter an indirect-fire MISTIC loiter station formation and all other sections of the unit are currently conducting attacks.

The processing of subroutine FRMLDR is almost identical to that of subroutine NATLDR. For details of the processing scheme, see Volume 2.

If ICE is not the leader as just discussed, then the route to join the unit formation will be a straight line from the current element's present position (XE, YE in common area /ICECOM/) to a point in space that is offset from the formation leader's position by the spacing specified in the unit formation data for section NSEC. The position of the formation leader is that which is projected for him at the end of ICE's present event. The processing sequence for determining the position of the terminal point in a transition route is as follows:

1. Call subroutine OFFSET (discussed earlier) to determine the formation spacing components DELX, DELY, DELZ for ICE relative to LDR.

2. Project the leader forward in space and time to the position he will occupy at the end of ICE's present event. Denote this position as XLD, YLD, ZLD.

3. Determine the leader's direction of motion, TDIR, at the projected position.

4. Compute coordinates of the terminal end of ICE's transition route by the relations

$$XO = XLD + DELX \sin(TDIR) + DELY \cos(TDIR)$$
$$YO = YLD - DELX \cos(TDIR) + DELY \sin(TDIR)$$
$$ZO = ZLD + DELZ.$$

In the above procedure, subroutine APFDYS is used to project the leader in time and space. This subroutine has several uses and was discussed in detail in reference 10. It is also used in the movement model of Chapter 7. The data XLD, YLD, ZLD are stored in common area /COPTER/ as follows:

$$XLD = COPTER(N + 1,7,1)$$
$$YLD = COPTER(N + 1,7,4)$$
$$ZLD = COPTER(N + 1,7,7)$$

where N is the number of aerial sections being represented. For a complete description of common area /COPTER/, see Volume 2.

The direction of motion of the leader at his projected position is computed as follows:

$$TDIR = \tan^{-1}\left(\frac{YSPD}{XSPD}\right)$$

where

$$XSPD = COPTER(N + 1,7,2)$$
$$YSPD = COPTER(N + 1,7,5).$$

These last entries in common area /COPTER/ are, as indicated, the X and Y components of the leader's projected speed at XLD, YLD, ZLD.

The battle time for which XLD, YLD, ZLD are computed is

$$TA = CLOCKT + EVHTIM(LWC)$$

where CLOCKT is the current element's clock time (common area /ICECOM/) and EVHTIM(LWC) is the input standard event time for a helicopter element with aerial weapon code LWC as defined previously.

The transition route is stored in the arrays XOPT, YOPT, ZOPT as are all routes. In this case, the arrays contain three valid points as will be discussed, so the length is stored as IKAP = 3 in common area /ICAP/. The first point is ICE's present position; i.e.,

$$XOPT(1) = XE$$
$$YOPT(1) = YE$$
$$ZOPT(1) = ELVATE(XE, YE, ICE).$$

Function ELVATE is described in Chapter 9 and Volume 2 and the value returned is the elevation of ICE above the zero elevation plane.

The terminal position of the transition route XOPT(3), YOPT(3), ZOPT(3) is adjusted slightly from the coordinates XO, YO, ZO discussed previously. The adjustment is included to allow some tolerance in the movement control process. The assumption is made that the transition route is complete whenever the current element reaches a position that is within fifty (50) meters of his proper formation position. This procedure allows the model to view a section as being in the unit formation whenever the section leader is sufficiently "close" to his proper formation position.

Thus, from the above discussion:

$$XOPT(3) = XO - RE \cos(ANG)$$
$$YOPT(3) = YO - RE \sin(ANG)$$
$$ZOPT(3) = ZO$$

where RE = 50 and

$$ANG = \tan^{-1}\left(\frac{YO - YE}{XO - XE}\right).$$

Note, however, that a situation might arise in which RE is not equal to fifty meters. This occurs when XO, YO is closer than fifty meters to XE, YE. When this situation arises, RE becomes the distance that exists.

The second point in the route XOPT(2), YOPT(2), ZOPT(2) is normally selected arbitrarily to be at a point one hundred (100) meters from XE, YE. The only time this is not true is when XO, YO is closer than 150 meters to XE, YE. Then, the point coincides with XE, YE, ZE. This second point in the

route is included to provide additional information about the route to the movement model. Thus, for XOPT(2), YOPT(2), we have the two sets of relations

$$XOPT(2) = XE + RB\cos(ANG)$$
$$YOPT(2) = YE + RB\sin(ANG)$$

where ANG is as defined previously and

$$RB = 0 \text{ if } \sqrt{(XO-XE)^2 + (YO-YE)^2} < 150$$

$$RB = 100 \text{ if otherwise.}$$

For ZOPT(2), we use the relation

$$ZOPT(2) = ZOPT(1) + \frac{RS}{R}(ZOPT(3) - ZOPT(1))$$

where

$$RS = \sqrt{(XOPT(2) - XOPT(1))^2 + (YOPT(2) - YOPT(1))^2}$$

and

$$R = \sqrt{(XOPT(3) - XOPT(1))^2 + (YOPT(3) - YOPT(1))^2} .$$

This computation places the second elevation point on a line from XOPT(1), YOPT(1), ZOPT(1) to XOPT(3), YOPT(3), ZOPT(3). Thus, the transition route accounts for the fact that the section may have to change elevations from its previous route in order to enter the unit formation.

When ICE is returning to a unit loiter station and no other sections are occupying the station (ICE = LDR), then special processing is required in subroutine RTJOIN. The procedure is quite simple, however. The terminal position XO, YO, ZO of the route is computed by the process described in the next paragraph. Then, processing is continued according to the procedures that were outlined above for the case in which XO, YO, ZO is a regular formation position.

In the special case being analyzed, the point XO, YO, ZO is chosen as a point on the recorded unit loiter station route. The point selected is the one in the unit route arrays that is closest to XE, YE. Therefore, the procedure is as follows:

1. Conduct a search of the unit loiter station route arrays XRT(I, MANUN), YRT(I, MANUN) where I = 1,...,ICAP1(MANUN).

2. Select that point ISAVE that minimizes

$$R = \sqrt{(XRT(I,MANUN)-XE)^2 + (YRT(I,MANUN)-YE)^2} .$$

3. Record

$$XO = XRT(ISAVE, MANUN)$$
$$YO = YRT(ISAVE, MANUN).$$

Actually, the search of the route arrays must extend only over four points since the route repeats itself. In subroutine RTJOIN, the points I = 10, 11, 12, 13 are analyzed.

The terminal point ZO is found from the recorded elevation of the loiter station route by the relation ZO = ZRT(10,NAT). Note that the tenth route point is arbitrarily selected since all the route points are at the same elevation above the zero elevation plane.

Generating a Section Formation Route

The final type of route that must be selected in subroutine PICKRT is one for a section that is flying in a unit formation. A value of nine for IRS triggers this type of route selection process and subroutine RTSECT is designed to perform the processing.

A route for a section in a unit formation is required whenever a section has successfully completed a transition route. The section is then considered to be a part of the unit formation and selects its route from the unit route arrays, properly offset to account for the section's position in the unit formation.

The number of points in the new route corresponds to the number of points recorded for the unit route, therefore, IKAP = ICAP1(MANUN). The offsets of the section in the unit formation relative to the formation leader are DELX, DELY, DELZ and are computed by subroutine OFFSET in a manner that has already been discussed. The procedure of subroutine RTSECT consists of loading the IKAP points in the arrays XOPT, YOPT, ZOPT from the arrays XRT, YRT, ZRT, accounting for the offsets DELX, DELY, DELZ.

For segment I of the maneuver unit route, bounded by the route points I and I + 1, the following equations apply for the offset route of the section

$$\begin{aligned} XOPT(J) &= X2 + DELX \sin(TDIR) + DELY \cos(TDIR) \\ YOPT(J) &= Y2 - DELX \cos(TDIR) + DELY \sin(TDIR) \\ ZOPT(J) &= Z2 + DELZ \end{aligned}$$

where

$$TDIR = \left\{ \begin{array}{l} \text{direction of motion along segment I,} \\ \tan^{-1}\left(\dfrac{Y1 - Y2}{X1 - X2}\right) \end{array} \right.$$

$$\begin{aligned} X1 &= XRT(I, MANUN) \\ Y1 &= YRT(I, MANUN) \\ Z1 &= ZRT(I, NAT) \\ X2 &= XRT(I+1, MANUN) \\ Y2 &= YRT(I+1, MANUN) \\ Z2 &= ZRT(I+1, NAT) \end{aligned}$$

and $J = IKAP - I + 2$.

Note, of course, that the arrays XOPT, YOPT, ZOPT are reversed from the arrays XRT, YRT, ZRT as previously specified. Note also that point J in the arrays XOPT, YOPT, ZOPT is associated with point I + 1 in the XRT, YRT, ZRT arrays. Thus, we need special relations to compute the point J = IKAP. The relations are:

$$\begin{aligned} XOPT(IKAP) &= XRT(1, MANUN) + DELX \cos(TDIR) + DELY \cos(TDIR) \\ YOPT(IKAP) &= YRT(1, MANUN) - DELX \cos(TDIR) + DELY \sin(TDIR) \\ ZOPT(IKAP) &= ZRT(1, NAT) + DELZ \end{aligned}$$

where

$$TDIR = \tan^{-1}\left(\frac{YRT(1, MANUN) - YRT(2, MANUN)}{XRT(1, MANUN) - XRT(2, MANUN)}\right).$$

Aerial Organizations and Formations

Throughout this chapter many references have been made to unit and section formations. When movement decision were formulated, either subroutine SECPRM or subroutine HFORM were referenced as setting the desired speed and formation pattern to be used during the new movement activity. Also,

when a route to be used by a section joining a unit formation was selected in subroutine RTJOIN, the desired position of the section leader within the unit formation was discussed. The calculation of this position was indicated as being performed in subroutine OFFSET. We now turn our attention to a discussion of these service subroutines.

Subroutine HFORM

Subroutine HFORM is used to determine the desired speed of an aerial unit, the desired speeds of the sections operating in the unit formation, and the formation pattern numbers to be used by the section(s), the platoon(s) (if appropriate), and the team (if appropriate) that comprise the unit. This subroutine must be used at the time that an aerial unit comes on to the battlefield and thereafter any time that a section within the unit joins or leaves the unit formation. The subroutine is used in both contexts in the movement controller of this chapter. Here, we discuss the processing accomplished by subroutine HFORM.

The desired speed and formation patterns to be used in an aerial unit NAT are dependent upon the activity being performed by the unit. As viewed by subroutine HFORM, the activities are indicated by the variable IFTN where:

$$\text{IFTN} = \begin{cases} 1 & \text{if NAT is performing enroute movement,} \\ 2 & \text{if NAT is loitering as a unit, and} \\ 3 & \text{if NAT is searching for a target as a unit.} \end{cases}$$

A value of one is appropriate any time NAT is moving across the battlefield to a new objective area. A value of two is appropriate any time the unit loiters as a single entity (on or off the battlefield while awaiting a mission, in a defensive position, at a retirement position, or delivering indirect-fire MISTIC support). Finally, a value of three is appropriate only when the unit is performing a counterbattery observation mission (see Chapter 1).

The desired speed of maneuver unit I is recorded as SPDMU(I). The correct value is determined from the input array HSPEED as follows:

$$\text{SPDMU(I)} = \text{HSPEED(IFTN, LWC)}$$

where LWC is the aerial unit weapon code as defined at the beginning of the chapter.

Now, the desired speed of each section operating in the unit formation is normally the desired speed of the maneuver unit. Therefore, the relation used to set the desired speed of the sections in the unit is SPDSE(NSEC) = SPDMU(I).

However, if section NSEC is joining the unit formation (JUNACT(NSEC) = 0, 1; JPHASE(NSEC) = 1) then, arbitrarily, the desired speed of NSEC is set as

$$\text{SPDSE(NSEC)} = 1.1\ \text{SPDMU(I)}.$$

This relation assures that sections attempting to enter the formation will do so as rapidly as possible while maintaining reasonably compatible speeds with respect to the rest of the unit's sections.

The formation pattern numbers are determined from the input array IHFNA and are recorded in the arrays FORMTE, FORMPT, and FORMSE. The procedures used are outlined below.

If the unit is a team (as defined in Chapter 1), the team number NT is determined and the formation pattern number is recorded by the relation FORMTE(NT) = IHFNA(IFTN, 4, LWC) where IFTN and LWC are as defined in a previous paragraph. If the unit is not a team (i.e., if the unit is a platoon or a section), no value is required for a team formation pattern.

After the team formation is determined (if required), the formation pattern number for each platoon NP in the unit is determined. If the unit is a platoon, only one such number is required; and if the unit is a section, no determination is made. The relation used is

$$\text{FORMPT(NP)} = \text{IHFNA(IFTN, 3, LWC)}$$

where IFTN and LWC are as previously defined.

Finally, the formation pattern number for each section NS in the unit is determined. However, only those sections NS that are actually flying in the unit formation at the time that the determination is made, are included. The reason for this convention is that sections operating independent of the unit have a separately recorded formation pattern number. Also, in determining formation spacings for elements flying in the unit formation, an accounting of sections missing from the unit formation is needed. For clarification of the above arguments, see the descriptions of subroutines OFFSET, ADJPOS and SECPRM that appear in subsequent paragraphs of this section.

To determine whether section NS is operating in the unit formation, the section formation pattern may be used. If FORMSE(NS) = 0, the section is not in the unit formation. This value is set prior to a call to subroutine HFORM if a section is leaving the unit. If the formation pattern number is not zero, then its proper value should be determined from the relation

$$\text{FORMSE(NS)} = \text{IHFNA(IFTN, I, LWC)}$$

where

$$I = \begin{cases} 1 & \text{if NS is in the first position of a platoon, and} \\ 2 & \text{if otherwise} \end{cases}$$

and where IFTN and LWC are as defined previously. If a section is joining the unit formation (JUNACT(NSEC) = 0, 1), then FORMSE(NS) is set to one prior to the call to subroutine HFORM. In this way, the section pattern number of a joining section will be set by the subroutine.

Subroutine SECPRM

Subroutine SECPRM is used to determine the desired speed and formation pattern to be used by an aerial section that is commencing independent movement.

The desired speed and formation pattern to be used by aerial section NSEC are dependent upon the activity being performed. As viewed by subroutine SECPRM, the activities are indicated by the variable IFTN where

$$\text{IFTN} = \begin{cases} 1 & \text{if NSEC is performing enroute movement,} \\ 2 & \text{if NSEC is loitering,} \\ 3 & \text{if NSEC is searching for targets, and} \\ 4 & \text{if NSEC is conducting attack route movement.} \end{cases}$$

A value of one is appropriate if NSEC is moving to a defensive or retirement position or if NSEC is moving to rejoin the unit in a new mission operations area. A value of two is appropriate any time NSEC is loitering (while waiting for fire support, in a defensive position or at a retirement position). A value of three is appropriate any time independent search is permitted (IUNACT(NAT) = 1, 2, 3, 4, 6 or 8 and JUNACT(NSEC) = 1) and finally a value of four is appropriate any time an attack route is being flown (JUNACT(NSEC)=2).

The desired speed of NSEC is recorded as SPDSE(NSEC). The correct value is determine from the input array SPEEDS as follows:

$$\text{SPDSE(NSEC)} = \text{SPEEDS(IFTN, LCOD)}$$

where LCOD is the aerial section weapon code that has been defined previously.

The formation pattern number is determined from the input array ISFNA and is recorded in the array HFORMS. The relation is

HFORMS(NSEC) = ISFNA(IFTN, LCOD)

where IFTN and LCOD are as defined previously.

Now, if the section NSEC is leaving the mission operations area (JUNACT(NSEC) > 3) because of the independent movement, then subroutine SECPRM also determines whether or not a new unit leader must be designated. This redesignation is required if the leader of NSEC has been operating as the maneuver unit leader; i.e., if MANLDR(MANUN) = ICE.

The processing required to determine the new leader is similar to that which was described in the discussion of subroutines NATLDR and FRMLDR. The formation hierarchy (see p.6-21) is searched for the section leader who appears first in the hierarchy and whose section is still operating in the unit mission area. This element is designated as MANLDR(MANUN).

Subroutine OFFSET

It is often important to know the proper position of an element within an aerial formation. For example, when an aerial unit initiates its first mission as discussed in this chapter, the position of the leader of the unit is known from initialized input data. However, the position for each following element in the unit formation must be computed. These computations are based upon the position and direction of motion of the leader, the formation positions of the followers and the formation patterns that are being used. As another example, the TAPCOM II movement model (Chapter 7) represents the movement of an aerial section leader with detailed equations of motion. Other elements within the section are simply placed at the positions they should occupy relative to the section leader.

Subroutine OFFSET (Volume 2) has been designed to compute the position of a given element with respect to a formation leader. This information is contained in the three quantities DELX, DELY and DELZ as defined in Figure 6.4. As may be noted, a standard right-hand Cartesian coordinate system is assumed, with origin at the formation leader's position and positive Y axis in the direction of travel.

The subroutine is designed to operate in any one of three modes, depending upon the input control quantity ICNT. If ICNT = + 1, the position of the leader of a section, relative to the maneuver unit leader, is determined. If ICNT = 0, the position of an element relative to the element's section leader

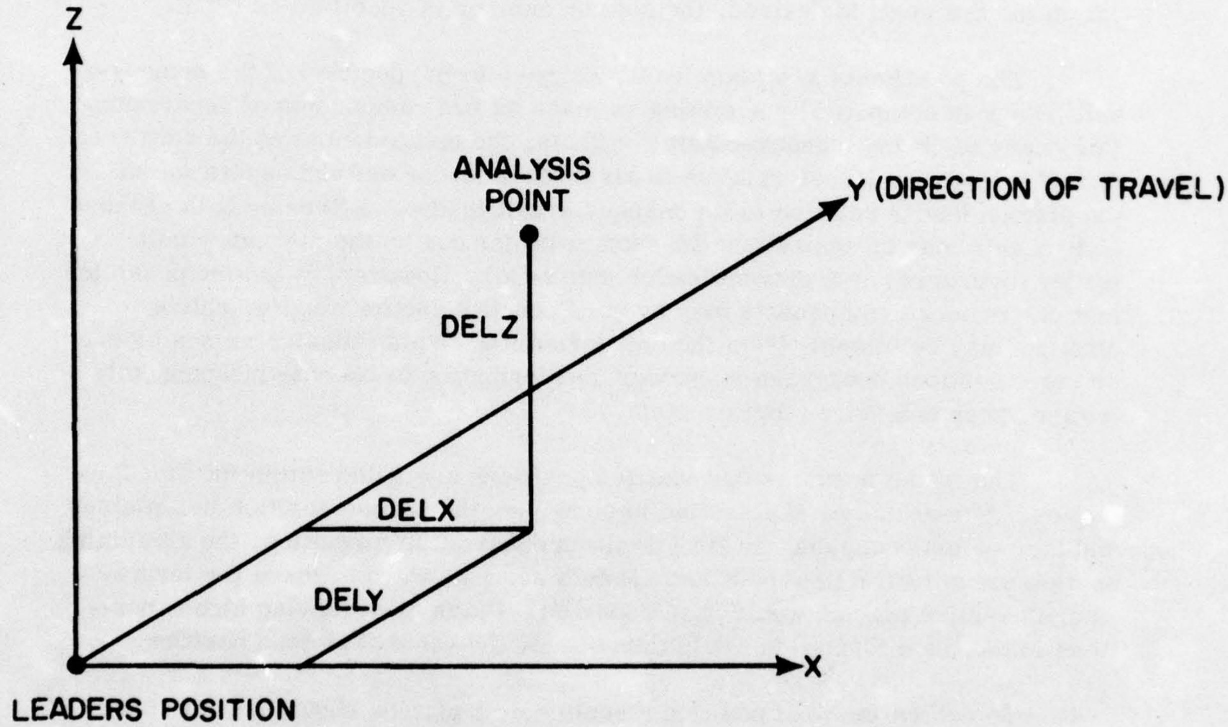


Figure 6.4.--Formation Position Coordinates

is determined. Finally, if ICNT = -1, the position of a specified section's center of observation relative to the section leader is determined. A section's center of observation is used in determining the principal direction of observation for elements in a section and is discussed in more detail in Chapter 5 and in reference 7.

In the cases in which the position of an element is desired (ICNT ≥ 0), the element number is specified as IHCE. This element can be the current element ICE or otherwise as required. In the case where the center of observation for a section is desired, the section number is specified as IHCE.

The position of a section leader relative to the position of the maneuver unit leader is computed by summing as many as two components of separation. For example, if the maneuver unit is a team, the components are the separation of the section leader relative to his platoon leader and the separation of the platoon leader relative to the maneuver unit leader. Either or both of these components may be zero since the section leader can be the maneuver unit leader (both zero) or a platoon leader (one zero). However, it is also possible that one or more components may be zero because sections and/or entire platoons may be missing from the unit formation. This situation arises as one or more sections depart the maneuver unit formation to operate independently (retire, seek defensive positions, etc.).

The model assumes that vacated positions are filled within the unit formation. For example, if a section is occupying the second position in a platoon and the section occupying the first position departs the formation, the remaining section occupies the first position. If both sections were to leave the formation, the entire platoon would have departed. Platoons occupying higher positions in the team formation would then occupy the vacated platoon position.

To determine what position a section or a platoon should occupy in a formation, subroutine ADJPOS (Volume 2) is used. First, externally we determine from input data the position JKPOS that a section or platoon should occupy in its parent organization, NPAR, given that all sections are present. Then, we call subroutine ADJPOS to examine the sections or platoons occupying positions I < JKPOS to determine whether or not these units are present. If NPAR is a team (as specified by the calling parameter ICNT = 1), we decrement JKPOS by one each time an entire platoon is found to be missing. If NPAR is a platoon (ICNT = 0), we decrement JKPOS by one if a section is missing. Of course, as discussed previously, a section I is missing if FORMSE(I) = 0, and a platoon is missing if all sections I in the platoon have FORMSE(I) = 0.

With the exception of this adjustment procedure, the method for computing the offsets of an element relative to his leader is exactly the same as that which has been reported in preceding descriptions of TAPCOM (see reference 8). We only summarize the data and computations below.

The input data that describe the formation organization are as follows:

ISORG(I, ISEC)	=	the element in position I of section ISEC (I = 1, . . . , 4 with I = 1 reserved for the section leader)
IPORG(I, NP)	=	the section in position I of platoon NP (I = 1, 2)
ITORG(I, NT)	=	the platoon in position I of team NT (I = 1, . . . , 7)
LPOS(I)	=	the position occupied by element I in his section formation
ISPOS(I)	=	the position occupied by section I in its platoon
IPPOS(I)	=	the position occupied by platoon I in its team
MANTYP(I)	=	the type of organization represented by maneuver unit I (1 - section, 2 - platoon, 3 - team)
MANORG(I)	=	organization number of maneuver unit I (team number if MANTYP(I) = 3, platoon number if MANTYP(I) = 2, section number if MANTYP(I) = 1)
FORMSX(I, J)	=	number of spacing increments in X between position I and position 1 in a formation utilizing pattern J
FORMSY(I, J)	=	Y analog of FORMSX(I, J)
FORMSZ(I, J)	=	Z analog of FORMSX(I, J)
FORMXS(I)	=	scale factor in X for a formation of type I (1 - section, 2 - platoon, 3 - team).
FORMYS(I)	=	Y analog of FORMXS(I)
FORMZS(I)	=	Z analog of FORMXS(I).

These data are used in the computation of the offsets DELX, DELY, DELZ by the following procedures:

Element I Relative to Leader of Parent Section ISEC

1. Determine position of element I in section ISEC; i.e., set IPOS = LPOS(I).

2. Determine formation pattern being used by section ISEC; i.e., IFORM = FORMSE(ISEC) or IFORM = HFORMS(NSEC).

3. DELX = FORMXS(1)*FORMSX(IPOS, IFORM)
   DELY = FORMYS(1)*FORMSY(IPOS, IFORM)
   DELZ = FORMZS(1)*FORMSZ(IPOS, IFORM).

Leader of Section ISEC Relative to Leader of Parent Platoon NPLAT (valid only for MANTYP(MANUN) > 1)

1. Determine the position of section ISEC in the platoon; i.e., set JPOS = ISPOS(ISEC).

2. Adjust the position of section ISEC to account for sections missing from the platoon; i.e., call subroutine ADJPOS for platoon NPLAT.

3. Determine formation pattern being used by platoon; i.e., set JFORM = FORMPT(NPLAT).

4. DELX = FORMXS(2)*FORMSX(JPOS, JFORM)
   DELY = FORMYS(2)*FORMSY(JPOS, JFORM)
   DELZ = FORMZS(2)*FORMSZ(JPOS, JFORM).

Leader of Platoon NPLAT Relative to Leader of Parent Team NTE (valid only for MANTYP(MANUN) = 3)

1. Determine the position of platoon NPLAT in the team; i.e., set KPOS = IPPOS(NPLAT).

2. Adjust the position of platoon NPLAT to account for platoons missing from the team; i.e., call subroutine ADJPOS for team NTE.

3. Determine formation pattern being used by team; i.e., set KFORM = FORMTE(NTE).

4. DELX = FORMXS(3)*FORMSX(KPOS,KFORM)
DELY = FORMYS(3)*FORMSY(KPOS,KFORM)
DELZ = FORMZS(3)*FORMSZ(KPOS,KFORM).

This concludes our discussion of aerial organizations and formations. We turn our attention now to some of the most important processing that occurs in the section movement and fire control model. The processing to be discussed determines whether or not a direct-fire target should be engaged by an aerial section, and if so, the allocation of fire to be made.

Aerial Vehicle Section Target Selection Model

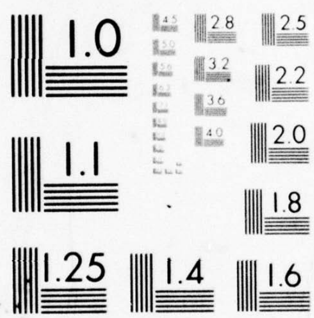
The reader will recall from previous discussions that the goal of an aerial unit performing a target of opportunity mission, a self-defense mission or a search-and-destroy mission, is to engage and defeat enemy weapons located within the mission operations area. To accomplish this goal, sections flying with the unit begin to search for targets immediately upon entering the mission area. This search operation continues until a target suitable for direct-fire attack is discovered, or until the section decides to leave the area. This latter decision may be made to allow the section to retire from the battlefield, to seek a protected defensive position or to commence a new mission with the unit. If the section decides to attack a target, it may resume searching for other targets upon completion of the attack or it may commence another attack upon the same target. Throughout these operations, the aerial section operates as a single tactical entity, with each element in the section occupying a specified position within the section formation.

The Aerial Vehicle Section Target Selection Model is designed to represent the decision process of an aerial vehicle section leader who is attempting to decide whether or not his section should commence an attack against enemy elements known to members of his section. This model is used each event that the decision to attack is available to the section leader. If the decision to attack is made, then the model formulates firing assignments for each member of the aerial section. These assignments are valid for the duration of the attack and specify which enemy element or elements are to be engaged and which weapons are to be employed by each section member. It is possible that one or more members of the section will be given no target assignments by the model, but these elements will continue to fly with the section. An assignment will not be made if a given aerial vehicle has insufficient ammunition of the types required for engagement of the enemy elements comprising the target.

AD-A040 052 OHIO STATE UNIV COLUMBUS SYSTEMS RESEARCH GROUP F/G 15/7
EXTENSIONS TO THE LAND COMBAT MODEL, (DYNCOM). VOLUME 1. HELICO--ETC(U)
DEC 71 G M CLARK DAAH01-70-C-0713
UNCLASSIFIED RF-2995-FR-71-1(U) NL

4 OF 4
AD
A040052

END
DATE
FILMED
6-77


1.0
4.5
5.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

The target selection model operates in four phases. The first phase consists of constructing a list of enemy elements known to members of the aerial section and of assigning priority weights to each of these elements. The priority weight assigned to an enemy element measures its significance as a target and is a function of several tactical considerations. One of the primary considerations is the nature of the target element itself. The model assumes that every ground element in the simulation can be classified as being either a heavy weapon element (e.g., a tank) or a light weapon element (e.g., an APC, crew-served weapon etc.). Therefore, the model employs this concept to differentiate between types of target elements. The nature of each ground weapon must be specified by input as follows:

$$\text{ITYPA(I)} = \begin{cases} 1 & \text{element I is a heavy weapon, and} \\ 0 & \text{element I is a light weapon.} \end{cases}$$

The use of this array will become clear in our discussion of target priority in a following paragraph.

The next computational phase of the target selection model determines the types of weapons currently available within the aerial section. A weapon becomes unavailable when its ammunition supply is depleted, therefore, weapon availability within a section can change after each target engagement.

The model assumes that aerial vehicle elements have available two classes of weapons as follows:

1. destructive (point) fire weapons, generally consisting of direct-fire missiles; and

2. suppressive (area) fire weapons, including such weapons as machine guns, rockets, etc.

Each aerial vehicle element may have up to six ammunition types available. The class of each weapon-ammunition category is specified as input by KAMOAV(IAM, IWC), where

$$\text{KAMOAV(IAM, IWC)} = \begin{cases} 1 & \text{destructive fire ammunition, and} \\ 0 & \text{suppressive fire ammuniton} \end{cases}$$

for ammunition code, IAM, and aerial vehicle weapon code, IWC.

The third computational phase of the target selection model determines a complex of one or more enemy weapons to be engaged by members of the aerial vehicle section. The procedure used to select the complex utilizes the list of known enemy weapons discussed earlier, along with their associated priority weights. The procedure also uses the weapon availability information constructed during the second computational phase. The best complex is that grouping of target elements that possesses the largest total priority weight among all complexes examined by the procedure, subject to the restriction that the selected complex can be attacked with the weapons available to the section. Details of the procedure will be given in a subsequent paragraph.

There are two situations that can exist in which no complex will be determined by the procedure outlined above. In the event that the list of known enemy elements is empty, no targets exist which can be attacked. In this case, the section leader will merely decide to continue searching for a suitable target. The other case occurs when the list is not empty but no complex can be determined that can be attacked with the available weapons. In this situation, the leader will decide to seek a protected defensive position since the targets that are known cannot be attacked.

The final computational procedure of the target selection model is used when a target complex can be specified for attack. During this phase, targets from the selected complex are assigned for engagement by members of the section. The weapon or weapons to be employed by each aerial vehicle are also specified.

We will now discuss each phase of the computational procedure in detail. This discussion should serve to clarify the general descriptions given in the preceding paragraphs.

Phase I--Priority Weight Determination

The initial phase of the target selection procedure is the determination of the priority weight, APRFCW(I), for each known enemy ground element, I. A list of all enemy ground elements known to elements of the aerial vehicle section is determined and stored in the array, LIST(I). Subroutine AIRPOR is then utilized to determine the total priority weight for each element in LIST(I) by considering nine priority weighting factors. The computational procedures of subroutine AIRPOR, along with the definition of the individual weighting factors utilized, is given below.

1. Set IASC = aerial vehicle section number
   IWC = aerial weapon code of section IASC

M = 1

J = LIST(M), the first ground enemy element number

APRFCW(J) = 0; J = 1,...,NLIST, the total number of known enemy ground elements

IAE(I) = ISORG(I, IASC); I = 1, 4, which are the element numbers of section IASC

CLONK = current clock time of the leader of IASC (element IAE(1))

TSUBM = duration of IAE(1)'s previous event.

2. If enemy element J is a heavy weapon; i.e., ITYPA(J) = 1, go to step 3. Otherwise, go to step 4.

3. Increment the total priority weight by ARFCW1(IWC), the weight specified for J being a heavy weapon; i.e., APRFCW(M) + ARFCW1(IWC) $\rightarrow$ APRFCW(M); go to step 5.

4. Increment the total priority weight by ARFCW2(IWC), the weight specified for J being a light weapon; i.e., APRFCW(M) + ARFCW2(IWC) $\rightarrow$ APRFCW(M).

5. If enemy element J fired during the previous event of IAE(1); i.e., CLONK - TSUBM $\leq$ EFELC(J), go to step 6. Otherwise, go to step 13.

6. Determine KZ, the target at which enemy element J fired; i.e., KZ = MDFAF(J).

7. If J fired at a ground element; i.e., LHICE(KZ) = 0, go to step 8. Otherwise, go to step 9.

8. Increment the total priority weight by ARFCW3(IWC), the weight specified for J having fired at a ground element; i.e., APRFCW(M) + ARFCW3(IWC) $\rightarrow$ APRFCW(M).

9. If J fired at an aerial vehicle element; i.e., LHICE(KZ) > 0, go to step 10. Otherwise, go to step 13.

10. If J fired at an element of section IASC; i.e., KZ = IAE(K); K = 1,... 4; go to step 12. Otherwise, go to step 11.

11. Increment the total priority weight by ARFCW4(IWC), the weight for J having fired at an aerial vehicle element not in section IASC; i.e.,

$$APRFCW(M) + ARFCW4(IWC) \rightarrow APRFCW(M);$$

go to step 13.

12. Increment the total priority weight by ARFCW5(IWC), the weight for J having fired at an element in section IASC; i.e., $APRFCW(M) + ARFCW5(IWC) \rightarrow APRFCW(M)$.

13. If any friendly ground elements are currently engaging enemy element J; i.e., J = MDFA F(K); K = 1,...,N, the number of friendly ground elements, go to step 14. Otherwise, go to step 15.

14. Increment the total priority weight by ARFCW6(IWC), the weight specified for J being engaged by a friendly ground element; i.e.,

$$APRFCW(M) + ARFCW6(IWC) \rightarrow APRFCW(M).$$

15. If any friendly aerial vehicle elements not in section IASC are currently engaging enemy element J; i.e., J = MDFA F(K); K = 1,...,NN, the number of friendly aerial vehicle elements not in IASC, go to step 16. Otherwise, go to step 17.

16. Increment the total priority weight by ARFCW7(IWC), the weight specified for J being engaged by aerial vehicle elements not in section IASC; i.e.,

$$APRFCW(M) + ARFCW7(IWC) \rightarrow APRFCW(M).$$

17. If any aerial vehicles in section IASC are currently engaging enemy element J; i.e., J = MDFA F(K); K = IAE(1),...,IAE(4); go to step 18. Otherwise, go to step 19.

18. Increment the total priority weight by ARFCW8(IWC), the weight specified for J being engaged by aerial vehicle elements in section IASC; i.e., APRFCW(M) + ARFCW8(IWC) → APRFCW(M).

19. If any element of IASC currently has full "eyeball" detection of J, go to step 20. Otherwise, go to step 21.

20. Increment the total priority weight by ARFCW9(IWC), the weight specified for any element of IASC having "eyeball" detection of enemy element J; i.e., APRFCW(M) + ARFCW9(IWC) → APRFCW(M).

21. If all known enemy elements have been considered; i.e., M = NLIST, go to step 23. Otherwise, go to step 22.

22. M + 1 → M; go to step 2.

23. The computations are completed.

Phase II--Aerial Section Weapon Availability

The second phase of the target selection procedure is to determine, for each element in the specified aerial vehicle section, the total number of destructive weapons, KAZ, and suppressive weapons, LAZ, currently available for employment by the aerial vehicle section. The assumption is made in the model that, during a single firing event, an aerial vehicle element can employ no more than one destructive weapon and no more than two suppressive fire weapons. The actual limitations on an aerial vehicle of weapon code IWC, are specified as input by KAMMAX(IWC), defined as follows:

$$\text{KAMMAX(IWC)} = \begin{cases} 0 & \text{vehicle can fire one destructive weapon only;} \\ 1 & \text{vehicle can fire one destructive and one suppressive weapon only;} \\ 2 & \text{vehicle can fire one destructive and two suppressive weapons only;} \\ 3 & \text{vehicle can fire one suppressive weapon only; and} \\ 4 & \text{vehicle can fire two suppressive weapons only.} \end{cases}$$

In order for a particular weapon-ammunition combination to be considered for employment by an aerial vehicle element, the current supply of that ammunition must meet the critical level, JAMOAV(IAM, IWC), specified as input for ammunition code IAM and aerial vehicle weapon code, IWC.

The computational procedures to determine the total number of destructive and suppressive fire weapons available to a specified aerial vehicle section, accomplished in subroutine AIRAMO, are given below.

1. Set IASC = aerial vehicle section number
   IAE(I) = element numbers in IASC; I = 1, . . . , 4
   KAZ = 0 the number of destructive weapons available to IASC
   LAZ = 0 the number of suppressive weapons available to IASC
   IWC = aerial vehicle section weapon code
   I = 1.

2. If no elements remain to be considered; i.e., IE = IAE(I) = 0, go to step 18. Otherwise, go to step 3.

3. Initialize the ammunition code, K; i.e., set K = 1.

4. Initialize the number of destructive weapons, KPOINT, and the number of suppressive weapons, KAREA, for element IE; i.e., set KPOINT = 0 and KAREA = 0.

5. Determine NAMO(K, IE), the current supply of ammunition K for element IE; i.e., call

   AMMO(IE, K, NAMO(K, IE)).

6. Determine AMOSPY(K, IE), the ratio of NAMO(K, IE) to the critical ammunition supply level, JAMOAV(K, IWC); i.e.,

$$AMOSPY(K, IE) = \frac{NAMO(K, IE)}{JAMOAV(K, IWC)} .$$

7. If NAMO(K, IE) is below the critical level; i.e., AMOSPY(K, IE) < 1, go to step 14. Otherwise, go to step 8.

8. If K is a destructive weapon; i.e., KAMOAV(K, IWC) = 1, go to step 9. Otherwise, go to step 12.

9. If IE can employ a destructive weapon; i.e., KAMMAX(IWC) $\leq$ 2, go to step 10. Otherwise, go to step 14.

10. If IE has already been considered for another destructive fire weapon; i.e., KPOINT = 1, go to step 14. Otherwise, go to step 11.

11. KAZ + 1 $\rightarrow$ KAZ
    KPOINT = 1; go to step 14.

12. If IE can employ a suppressive fire weapon <u>and</u> all possible ones have not been considered; i.e., KAREA = 0 <u>and</u> KAMMAX(IWC) = 0 <u>or</u> KAREA = 1 <u>and</u> KAMMAX(IWC) = 2 or 4; go to step 13. Otherwise, go to step 14.

13. LAZ + 1 $\rightarrow$ LAZ
    KAREA + 1 $\rightarrow$ KAREA.

14. If all ammunition codes have been considered for element IE; i.e., K = 6, go to step 16. Otherwise, go to step 15.

15. K + 1 $\rightarrow$ K; go to step 5.

16. If all elements of IASC have been considered; i.e., I = 4, go to step 18. Otherwise, go to step 17.

17. I + 1 $\rightarrow$ I; go to step 2.

18. The computations are completed.

<u>Phase III--Determination of Enemy Target Complex</u>

The third phase of the target selection procedure is to determine the best complex of enemy ground targets for engagement by the aerial vehicle section, the computations for which are accomplished in subroutine AIRFIR, described in a subsequent paragraph.

The maximum allowable radius of the circle utilized to describe the boundaries within which a ground target complex will be attacked is specified by input data as RADMAX(K, L) for an aerial vehicle section having currently available (K-1) suppressive fire weapons and (L-1) destructive fire weapons. The best complex of known enemy ground targets is determined by procedures

described in reference 9 utilizing subroutine TARCEN, whose flow chart is given in reference 4. Basically, the procedure used is to compute the centroid of enemy priority weights and to check if all elements under consideration lie within the target-complex circle centered at the centroid. If not, the element most distant from the circle is dropped from further consideration. The centroid for the remaining elements is then computed, and the process is repeated until all elements under consideration lie within the target complex circle. The enemy elements in the selected target complex are returned from subroutine TARCEN in LIST(I), I = 1,...,NLIST2, where NLIST2 is the total number of enemy elements in the selected complex.

Next, the number of heavy weapons, noted by IAA, and the number of light weapons, noted by IBB, are determined for the selected complex by subroutine AIRFIR. The desirability of the selected complex is then determined from DESAIR(IA, IB, KA, LA), specified by input data, defined as follows:

DESAIR(IA, IB, KA, LA) = the desirability factor for an aerial vehicle section utilizing (KA-1) suppressive fire weapons and (LA-1) destructive fire weapons against an enemy complex consisting of (IA-1) heavy weapons and (IB-1) light weapons.

Note that the desirability factor must be specified as equal to -1 for those combination of IA, IB, KA and LA which are to be disallowed.

In order to assure that the most desirable weapon configuration for the aerial section is determined relative to the selected target complex, the maximum value of DESAIR(IA, IB, KA, LA) is determined over KA = 1,..., KAZ + 1, and LA = 1,..., LAZ + 1. That is, the most desirable combination of the number of destructive fire weapons (KAZ) and suppressive fire weapons (LAZ) for the selected complex is determined and recorded.

If it is determined that no allowable configuration was found for the specified complex radius, RADMAX(KA, LA), the entire process is repeated for a smaller radius, RAD, computed as follows:

$$RAD = RADMAX(KA, LA) - RADINC(KA, LA)$$

where RADINC(KA, LA), specified by input data, is the incremental decrease in the allowable complex radius for determination of a reduced enemy complex. This process is sequentially repeated until either a desirable complex is found or the radius of the complex, RAD, has been reduced to a value less than or equal to zero. Note that only a complex of radius RADMAX(KA, LA) will be considered if RADINC(KA, LA) ≥ RADMAX(KA, LA) is specified by input.

The computations described above are accomplished in subroutine AIRFIR, the computational procedure of which is given below.

1. Set IASC = aerial vehicle section number
   IAE(I) = I^{th} element of section IASC; I = 1, . . . , 4
   NLIST = 0
   IWC = aerial vehicle section weapon code
   (XICE, YICE) = X, Y coordinates of the leader of section IASC (i.e., IAE(1))

$$\text{ICALL} = \begin{cases} 0 & \text{computations are for section, and} \\ 1 & \text{computations are for unit.} \end{cases}$$

2. Determine LIST(K); K = 1, . . . , NLIST, the enemy ground element numbers of those elements for which any element of section IASC has knowledge, by a call to subroutine GETDET.

3. If no element of section IASC has knowledge of any enemy ground element, go to step 4. Otherwise, go to step 5.

4. Record the fact that the section should continue searching for targets; i.e., set ITGASN = 1; go to step 33.

5. Determine APRFCW(I); I = 1, . . . , NLIST, the priority weight for each known enemy ground element, by a call to subroutine AIRPOR.

6. Determine KAZ and LAZ, the number of destructive and suppressive fire weapons, respectively, available to aerial vehicle section, IASC, by a call to subroutine AIRAMO.

7. Set KA = KAZ + 1
   LA = LAZ + 1.

8. Set DESMU = -1
   RAD = RADMAX(KA, LA).

9. If RAD, the specified target complex radius, is nonpositive; i.e., RAD $\leq$ 0, go to step 27. Otherwise, go to step 10.

10. Determine LIST(I); I = 1, . . . , NLIST2, the enemy elements in the selected complex; ASUM, the total complex priority weight; and (XC, YC), the coordinates of the complex centroid, by a call to subroutine TARCEN.

11. If the computations are for an aerial vehicle unit; i.e., ICALL = 1, go to step 33. Otherwise, go to step 12.

12. Set K = 1
    IAA = 0
    IBB = 0.

13. Set KCE = LIST(K), the K^{th} enemy element number in the selected complex.

14. If enemy element KCE is a heavy weapon; i.e., ITYPA(KCE) = 1, go to step 15. Otherwise, go to step 16.

15. Increment IAA, the number of heavy weapons in the complex, and IA, the subscript associated with IAA; i.e., IAA + 1 → IAA, IA = IAA + 1; go to step 17.

16. Increment IBB, the number of light weapons in the complex, and IB, the subscript associated with IBB; i.e., IBB + 1 → IBB, IB = IBB + 1.

17. If all enemy elements in the complex have been considered; i.e., K ≥ NLIST2, go to step 19. Otherwise, go to step 18.

18. K + 1 → K; go to step 13.

19. If the complex has the maximum desirability for those weapon combinations considered; i.e.,

    DESAIR(IA, IB, KA, LA) > DESMU,

    go to step 20. Otherwise, go to step 21.

20. Record information for the most desirable complex found so far; i.e., set

    DESMU = DESAIR(IA, IB, KA, LA)
    KAU = KA - 1
    LAU = LA - 1
    MU = M
    XCU = XC
    YCU = YC
    NLISTU = NLIST2
    ASUMU = ASUM

LISTU(KZ) = LIST(KZ); KZ = 1,...,NLIST2
IAAU = IAA
IBBU = IBB.

21. If KA $\leq$ 1, go to step 23. Otherwise, go to step 22.

22. KA - 1 $\rightarrow$ KA; go to step 19.

23. Set KA = KAZ + 1.

24. If LA $\leq$ 1, go to step 26. Otherwise, go to step 25.

25. LA - 1 $\rightarrow$ LA; go to step 19.

26. Set LA = LAZ + 1.

27. If a desirable complex was found; i.e., DESMU > 0, go to step 31. Otherwise, go to step 28.

28. Decrement the complex radius, RAD; i.e., RAD - RADINC(KA, LA) $\rightarrow$ RAD.

29. If RAD $\leq$ 0, go to step 30. Otherwise, go to step 9.

30. Record the fact that no desirable target complex was found and that the section should seek a defensive position; i.e., set ITGASN = 0; go to step 33.

31. Determine KELAGN(I, LHCE), the enemy element number assigned to aerial vehicle LHCE for gun system code I, and KAMAVL(I, LHCE), the ammunition code number to be employed by aerial vehicle LHCE for gun system code I, where

$$I = \begin{cases} 1 & \text{destructive weapon assignment,} \\ 2 & \text{first suppressive weapon assignment, and} \\ 3 & \text{second suppressive weapon assignment} \end{cases}$$

by a call to subroutine WASAIR.

32. Record the fact that a target assignment was made; i.e., set ITGASN = 2.

33. The computations are completed.

Phase IV--Target Assignment Procedures

The final phase of the target selection procedure is to determine the specific assignments of the aerial vehicle element-ammunition code combinations to specified enemy elements in the target complex which was selected. These procedures are accomplished in subroutine WASAIR, which will be described later in this section.

The following conventions regarding specific assignments are utilized in the model.

1. No aerial vehicle element may fire more than one destructive fire and two suppressive fire weapons during a single firing event.

2. Assign one destructive fire weapon of the aerial vehicle section to each heavy weapon in the selected target complex, if possible.

3. If more destructive fire weapons are available to the aerial section than there are heavy weapon targets in the complex, the best[1] destructive weapon assignment is made to each heavy weapon, and the remaining destructive fire weapons are not assigned for the current firing event.

4. If more heavy weapons exist in the target complex than available destructive fire weapons in the aerial vehicle section, assignments are made for the available destructive fire weapons to the most desirable heavy weapons in the complex. The remaining heavy weapons are assigned to the most desirable suppressive fire weapons

---

[1]The best assignment is determined by utilization of KAMPRD(I, J, L), the ammunition priority list for an aerial vehicle of weapon code L, ammunition code, J, and target type I, specified by input data as a largest integer rank of priorities, where

$$I = \begin{cases} 1 & \text{heavy target, and} \\ 2 & \text{light target.} \end{cases}$$

in the aerial section, if possible.[1] In other words, it is considered more important to assign aerial weapons against heavy targets than light targets, if desirable ammunition is available.

5. After all possible destructive weapon assignments have been made, and remaining heavy targets have been assigned to appropriate suppressive fire weapons, assignments of available suppressive fire weapons for the aerial section to light weapons in the target complex are made. Only those elements in the aerial vehicle section which did not receive a destructive weapon assignment are considered in this phase of target assignment.

6. For those aerial vehicle elements which received a destructive weapon assignment against a heavy target, up to two suppressive fire assignments may be made for each element. Because it is assumed that suppressive fire may not occur during the firing of a destructive fire weapon, suppressive fire must be delivered either before and/or after the firing of the destructive weapon. The limitation on these suppressive fire assignments are specified as input by KAMMAX(IWC)[2] previously defined, and the three possible cases are described below.

   a. If KAMMAX(IWC) = 0, no suppressive fire assignment is possible.

   b. If KAMMAX(IWC) = 1, one suppressive fire weapon may be assigned, in addition to the destructive weapon assignment. It is assumed that this suppressive fire weapon will be fired after the firing of the destructive weapon. Furthermore, this firing may occur on a different target than the one selected for destructive fire (if such an assignment is possible).

---

[1] The list of enemy elements in the selected target complex are ordered so that the heavy weapons appear at the top of the list, followed by the light weapons in the complex.

[2] IWC is the weapon code of the aerial vehicle element being considered.

c. If KAMMAX(IWC) = 2, two suppressive fire weapons may be assigned, in addition to the destructive weapon assignment. In this case, a suppressive fire assignment prior to the destructive weapon firing is first attempted. The target assigned in this case is always the destructive fire weapon target, since it is assumed that prior to the firing of a destructive weapon, no other target considerations are possible.

If a suppressive weapon assignment is made for firing prior to destructive weapon firing, the following information is recorded in FACTL(NZ, I) for aerial vehicle I; where

$$NZ = \begin{cases} 1 & \text{the number of rounds per burst,} \\ 2 & \text{the number of bursts to be fired,} \\ 3 & \text{the time to fire one burst,} \\ 4 & \text{the time between bursts, and} \\ 5 & \text{the total time required to fire all bursts.} \end{cases}$$

The above information is determined utilizing the input array, BRAIR(I, J, N) for aerial vehicle weapon code J, employing ammunition type I, firing data type N, where

$$N = \begin{cases} 1 & \text{number of rounds in one burst,} \\ 2 & \text{desired number of bursts per firing event,} \\ 3 & \text{firing rate (round/second), and} \\ 4 & \text{time between firing bursts.} \end{cases}$$

Whether or not a suppressive fire weapon assignment was made prior to the destructive weapon firing, an attempt to make a suppressive fire weapon assignment after the destructive weapon firing is made as described in case b above.

The firing assignments made as a result of the procedures described above are recorded in the arrays KELAGN(I, LHCE) and KAMAVL(I LHCE), defined below.

KELAGN(I, LHCE) = the enemy element number assigned to weapon designator I of aerial vehicle, LHCE; and

KAMAVL(I, LHCE) = the ammunition code to be employed in the attack of enemy element KELAGN(I, LHCE) for aerial vehicle LHCE employing weapon code I, where

$$I = \begin{cases} 1 & \text{destructive fire weapon,} \\ 2 & \text{first (or posterior) suppressive fire weapon, and} \\ 3 & \text{second (or prior) suppressive fire weapon.} \end{cases}$$

The targeting data defined above are converted into a more usable form in subroutine CONVRT. That is, the arrays IHTARG and IHAMO defined in a previous discussion of this chapter are constructed from the arrays KELAGN and KAMAVL, respectively. Table 6.13 indicates the associations that are made. These associations are based upon the definitions that have been given for the four arrays and upon the possible assignments that can be made by subroutine WASAIR.

Table 6.13

Firing Data Conversion

Assignment Possibility Number	IHTARG(I, L) IHAMO(I, L)				
	I				
	1	2	3	4	5
1	KELAGN(3, L) KAMAVL(3, L)			KELAGN(2, L) KAMAVL(2, L)	KELAGN(1, L) KAMAVL(1, L)
2				KELAGN(2, L) KAMAVL(2, L)	KELAGN(1, L) KAMAVL(1, L)
3	KELAGN(3, L) KAMAVL(3, L)				KELAGN(1, L) KAMAVL(1, L)
4					KELAGN(1, L) KAMAVL(1, L)
5	KELAGN(2, L) KAMAVL(2, L)	KELAGN(2, L) KAMAVL(2, L)			
6	KELAGN(2, L) KAMAVL(2, L)	KELAGN(2, L) KAMAVL(2, L)	KELAGN(3, L) KAMAVL(3, L)	KELAGN(3, L) KAMAVL(3, L)	

L = helicopter number

VARIABLE DEFINITION INDEX

Variable	Definition	Variable	Definition
AMOSPY	p. 6-117	FORMTE	p. 6-104
APRFCW	6-113	FORMXS	6-109 (input)
ARFCW1	6-114 (input)	FORMYS	6-109 (input)
ARFCW2	6-114 (input)	FORMZS	6-109 (input
ARFCW3	6-114 (input)	HALTDS	6-71, 75, 85, 93, 94 (input)
ARFCW4	6-115 (input)	HALTDU	6-70, 75, 85 (input)
ARFCW5	6-115 (input)	HFORMS	6-106
ARFCW6	6-115 (input)	HSPEED	6-103 (input)
ARFCW7	6-115 (input)	HXRT	6-64
ARFCW8	6-116 (input)	HYRT	6-64
ARFCW9	6-116 (input)	HZRT	6-64
ATDIR	6-92 (input)	IAE	6-114
ATLIM	6-92 (input)	IASC	6-113
BRAIR	6-125 (input)	ICAP	6-73
CBDET	6-63 (input	ICAP1	6-65
CF	6-34 (input)	ICAP2	6-65
CFUEL	6-40 (input)	ICE	6-2
CLOCKT	6-22	IDPC	6-16
CLONK	6-114	IFBMIS	6-31, 59
COPTER	6-98	IFIN	6-82 (input)
DELX	6-107	IFMC	6-33 (input)
DELY	6-107	IFRFL	6-30
DELZ	6-107	IFTN	6-45
DESAIR	6-119 (input)	IHAMO	6-52, 127
DIRMU	6-25	IHDFMC	6-55, 90 (input)
ECLOCK	6-30	IHFNA	6-104 (input)
EDIR	6-26	IHTARG	6-52, 127
EFELC	6-114	IKAP	6-66
ELOCX	6-27 (input)[1]	IMIST	6-30 (input)[2]
ELOCY	6-27 (input)[1]	INART	6-32 (input)[2]
ELOCZ	6-27	IPHASE	6-6 (input)
ESPD	6-26	IPORG	6-109 (input)
ETIM	6-62	IPPOS	6-109 (input)
EVHTIM	6-58 (input)	IRET	6-40
FACTL	6-125	IRS	6-13
FORMPT	6-104	IRTSIZ	6-65 (input)
FORMSE	6-8, 105	ISACT	6-6
FORMSX	6-109 (input)	ISEC	6-2
FORMSY	6-109 (input)	ISFNA	6-106 (input)
FORMSZ	6-109 (input)	ISORG	6-109, 3 (input)

XLSTAS	p. 6-71	(input)
XLSTAU	6-70	(input)
XMAX	6-75	(input)
XOPT	6-64	
XRT	6-65	
XS	6-43	
XSAVE	6-95	
YA	6-73	
YD	6-25,94	
YDFO	6-94	
YE	6-72	
YMAX	6-75	(input)
YOPT	6-64	
YRT	6-65	
YS	6-43	
YSAVE	6-95	
ZA	6-73	
ZOPT	6-64	
ZRT	6-65	
ZSAVE	6-95	

---

[1] For aerial vehicles, only the data for maneuver unit leaders need be initialized.

[2] See Table 6.9, p. 6-32.

REFERENCES

1. Hutcherson, D. C., "Fire Support Target Selection and Coordination," Chapter 2 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

2. Hutcherson, D. C., "Fire Request Radio Net Communications," Chapter 3 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

3. The Land Combat Model, DYNCOM Programmer's Manual, edited by G. Clark, et al., RF 2376 FR 70-4A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, April 1971, AD 872508.

4. The Land Combat Model, DYNCOM Programmer's Manual, edited by G. Clark, et al., RF 2376 FR 70-4B (U), Systems Research Group, The Ohio State University, Columbus, Ohio, April 1971, AD 872508.

5. Clark, G. M., "Movement Controller," Chapter 5 (in) The Tank Weapon System, edited by A. Bishop and G. Clark, RF 573 FR 69-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1969, AD 864919.

6. Selby, S. M., (ed.) Standard Mathematical Tables, The Chemical Rubber Co. (CRC), Cleveland, Ohio, 1965.

7. Clark, G. M., "Intelligence," Chapter 4 (in) The Tank Weapon System, edited by A. Bishop and G. Clark, RF 573, FR 69-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1969, AD 864919.

8. The Land Combat Model, The Aerial Platform Combat Operations Module, edited by G. Clark and D. Hutcherson, RF 2376 FR 71-3 (U), Systems Research Group, The Ohio State University, May 1971, AD 887264.

9. Demmy, S., "Indirect-Fire Ballistic Weapons," Chapter 10 (in) The Tank Weapon System, edited by A. Bishop and G. Clark, RF 573 FR 69-2B (U), Systems Research Group, The Ohio State University, September 1969, AD 864920.

10. Rheinfrank, J. J. III, and C. B. McCartney, "Aerial Platform Flight Dynamics Simulator (APFDYS)," Chapter 7 (in) Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, edited by G. M. Clark, et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

CHAPTER 7

HELICOPTER ELEMENT MOVEMENT

by
D. C. Hutcherson

Introduction

Helicopter element movement during its current event is simulated by subroutine HELMOV whose flow chart appears in Volume 2. Subroutine HELMOV is called from the DYNCOM main program after all movement and firing decisions have been made for a helicopter element and before any of the firing activities are simulated.

The model assumes that the following state variables may change because of movement:

1. battlefield position coordinates of the helicopter,
2. altitude of the helicopter,
3. direction of heading for the helicopter,
4. average speed of the helicopter,
5. fuel remaining, and
6. final speed of the helicopter.

At the time subroutine HELMOV is called, the movement state variables of a helicopter are as recorded at the end of the helicopter's previous event. After processing, the values of the state variables are those that exist at the end of the current event. The model is constructed so that movement is either for a specified time or for a specified distance. In the first case, the duration of the current event is computed and used by HELMOV to predict movement; while in the second case, the time required to move the specified distance is computed. Each helicopter element being represented is processed by the model when the element becomes current.

In addition to changes in a helicopter's movement state variables, there are also DYNCOM movement indicators that change because of movement, and new values for these indicators must also be recorded. Therefore, subroutine HELMOV records new values for the flags that indicate:

1. whether or not the element achieved its desired position during the event,

2. whether or not the element achieved the initial position of a firing run during the event,

3. whether or not the element is currently conducting a firing run,

4. the current position of the element within route arrays recorded for the element.

Processing required within subroutine HELMOV is different depending upon whether or not the element being processed is the leader of a section. The model assumes that elements within a section always operate in a specified formation regardless of the type of movement being performed by the section. Thus, the follower elements can be assumed always to occupy specified positions within the formation and to always fly at the offset from the leader appropriate to the formation position occupied. If the leader of the section is always processed first so that his position is known at the beginning of a follower's event, the follower can then be moved for the same period of time that the leader was moved and its position can simply be the position it should achieve in order to occupy the proper formation position. Its speed, direction and fuel remaining can be recorded as those possessed by the leader, and the DYNCOM movement indicators can be the same as the leader's. This approach is, in fact, taken by subroutine HELMOV. The reader will recall from Chapter 1 that the leader is, indeed, processed first so the approach is valid.

Subroutine HELMOV Processing

In the paragraphs that follow, we will describe the processing of subroutine HELMOV. The first discussion is devoted to leader processing while subsequent discussion will be devoted to follower processing.

Leader Processing

The element being processed is the current element ICE. This element is the leader of section ISEC and a member of maneuver unit MANUN as specified in common area /ICECOM/. The aerial section number NSEC and the aerial unit number NAT are specified as before:

NSEC = NAVSEC(ISEC); and

NAT = MANHEL(MANUN).

The weapon code of the current element is KWCOD as specified in common

area /ICECOM/, and the weapon code of the aerial section can be determined as discussed in previous chapters; i.e.,

LWC = KWCOD - MAXLWC

where MAXLWC is specified in common area /NUMBER/.

Retired Sections

For a lead element, the first processing required is concerned with special situations that arise when the section has retired from the battlefield. The reader will recall from Chapter 1 that helicopter sections may retire from the battlefield when fuel or ammunition supplies are depleted or when the section suffers sufficient attrition. The model that treats these decisions is discussed in Chapter 4, while the model that implements the decisions is discussed in Chapter 6. Here, we are concerned only with processing that must be accomplished when retiring sections or units reach their retirement positions. The model has been constructed so that these sections are removed from the battle. Hence, no movement processing is required. Note, however, that they are not removed until they have completed the cross-country movement required for retirement.

To remove a section from the battle, all that is required is to set the clocks of the elements to large positive numbers again. Thus, when the leader is found to have reached his retirement position, his event time TIME is set to a large value, his desired position code IDPC is set to one to indicate he reached his desired position and processing is complete. Then, when follower elements become current, they are removed one by one from the battle since their event times assume the value of the leader's last event time.

To determine whether a section has reached a retirement position, the movement activity flags for helicopter sections and units are used. The reader will recall that section NSEC is retiring independent of the rest of its unit if JUNACT(NSEC) = 4. The reader will also recall that a section that has been over the battlefield is retiring with its unit NAT if

JUNACT(NSEC) = 0 and

IUNACT(NAT) = 10.

If either of the above conditions hold, all that remains to be determined is whether the section has completed its enroute movement. This, of course, is indicated by JPHASE(NSEC) = 0.

Nonretired Sections

If it is determined that the section has not reached a retirement position, movement for the lead element must be simulated. The first task is to determine the event time TIME to be used for movement.

The model is constructed so that the event time is either specified before subroutine HELMOV is called or must be calculated by HELMOV. The former case arises when the movement and fire controller of Chapter 6 has made a decision which permitted calculation of the event time. In this case, TIME will be positive and specifies the duration of the movement event; thus, helicopter movement during the event is a variable which is calculated by HELMOV based upon TIME. In the second case, TIME must either be computed so that movement can be predicted as in case 1, or TIME must be predicted as the amount of time required to move a specified distance.

The only instance in TAPCOM II when time required to move a specified distance is computed occurs when a helicopter section is performing the first phase of an attack as discussed in Chapter 6. The reader will recall that the route specified for the section is only long enough to permit the first attack phase to be conducted in this case. The event time is computed by the procedure to be described when we discuss the aerial platform flight dynamics simulator in a following section.

If the event time is unknown (TIME = 0) and if the section is not conducting the first phase of an attack, the event time must be computed. As modeled, the event time is determined by the relation

$$\text{TIME} = \text{EVHTIM(LWC)}$$

where the EVHTIM array is input and is contained in common area /EVHTIM/ and where LWC is the weapon code of the aerial section.

With the event time determined, the aerial platform flight dynamics simulator can be used to predict movement. This model was reported in reference 1 and is implemented in subroutine APFDYS whose flow chart appears in Volume 2. We will discuss here only the modes of operation of the model and input/output data.

Subroutine APFDYS uses as input data the variables TIME, ICNT and IDPC where TIME has been discussed before. Table 7.1 summarizes the modes of operation of APFDYS based on the above input data.

Table 7.1

Subroutine APFDYS Modes of Operation

TIME	ICNT	IDPC	Mode of Operation
T	0	0	Move current element for time T and set IDPC = 1 if ICE reaches the end of the specified route
0	1	0	Move current element until the end of the specified route is reached. Return the time required as T and set IDPC = 1
T	N	0	Extrapolate the position of element N to battle clock time T where N $\neq$ ICE

The first two modes of operation in Table 7.1 are used in subroutine HELMOV, as they are concerned with movement of the current element. The parameter IDPC is the desired position code of the current element and is always input as zero. It is returned as a one if the current element reaches the end of its route. In these modes, the movement state variables of the current element will be changed and the DYNCOM movement indicators will be set as required.

In the last mode of Table 7.1, subroutine APFDYS merely extrapolates the element N for a period of time determined from T but does not change any of the movement state variables for element N, nor do any of the movement indicators change. This mode is used in applications where it is desired to know the position of some aerial element at some time other than his currently recorded clock time. Such situations arise in the air defense models reported in reference 2 and in the model implemented by subroutine RTJOIN reported in Chapter 6.

In modes one and two, the state variables that change and are recorded by subroutine APFDYS are as follows:

XE,YE = position coordinates of the current element ICE (common area /ICECOM/)

ELOCZ(IHCE) = altitude of the current element above the terrain where IHCE is the helicopter number corresponding to ICE (common area /ELOCZ/)

ESPD(ICE) = final speed of ICE (common area /ESPD/)

DIR = direction of heading of ICE (common area /ICECOM/)

SPD = average speed of ICE (common area /ICECOM/).

The last three variables ESPD(ICE), SPD and DIR are measured in a plane parallel to the battlefield XY plane.

The movement indicators that change are the desired position code IDPC, and the route arrays position indicator LCPE(ICE). The first of these indicators was defined in our discussion of Table 7.1 while the latter merely contains the subscript of the point in the route arrays to which the current element is going. The variable is recorded in common area /LCPE/.

After processing by subroutine APFDYS, the remaining fuel supply of the elements in the section must be revised and values for the movement indicators not set by subroutine APFDYS must be determined.

The fuel supply of each member in section NSEC is WFUEL(NSEC) and is recorded in common area /WFUEL/. Obviously, these data must be initialized. The model assumes that fuel is used at the input rate RFUEL(I,LWC) where LWC is as defined previously and I is defined as

$$I = \begin{cases} 1 & \text{if NSEC is loitering} \\ 2 & \text{if NSEC is performing enroute movement} \\ 3 & \text{if NSEC is attacking a target} \\ 4 & \text{if NSEC is searching for a target.} \end{cases}$$

The array RFUEL is contained in common area /RFUEL/.

Since each element in NSEC is assumed to possess exactly the same characteristics and to always operate with NSEC, it is sufficient to record only one value for section NSEC. This value is recorded when the leader of NSEC is processed.

Finally, the movement indicators IFPC in common area /ICECOM/ and MSFP(ICE) in common area /MSFP/ must be determined for the current element ICE. The first variable is the fire position code and is defined as follows:

$$\text{IFPC} = \begin{cases} 1 & \text{if ICE is in a fire position} \\ 0 & \text{if otherwise.} \end{cases}$$

For helicopters, being in a fire position is equivalent to conducting a firing run. Thus, IFPC is set to one when ICE reaches the initial point of an attack and is not reset until the attack is terminated, either after the first or the second phase of the attack.

The variable MSFP(ICE) is set to one if a fire position is achieved during an event and is zero at all other times. The variable is used to determine when fire is to commence. Thus, MSFP(ICE) is set to one when IFPC transitions from zero to one and is set to zero at all other times.

Follower Processing

As explained previously, when a follower is processed by subroutine HELMOV, the movement state variables and indicators are already known for the leader. All that is required is conversion of data recorded for the leader into data to be recorded for the follower.

For follower element ICE belonging to section ISEC and maneuver unit MANUN, the leader of the section is LDR = ISORG(1,ISEC). The array ISORG is contained in common area /ISORG/ and LDR is the element in position one of section ISEC's formation.

The event time and some of the movement state variables and indicators for ICE are determined as outlined in Table 7.2. The position of ICE is determined by the procedure to be outlined in the next paragraph.

Current Element Position

The position of the current element at the end of the event is to be at the offsets from the section leader specified by the formation pattern being used by the section, and by the position of the current element in this formation. The reader will recall from Chapter 6 that subroutine OFFSET is designed to yield these offset data. The offset behind or in front of the leader is DELY, the offset to the left or right is DELX and the offset above or below is DELZ. Thus, subroutine HELMOV uses subroutine OFFSET to define these values.

Table 7.2

State Variables for an Aerial Element

Variable	Description	Source
DIR	Direction of heading (common area /ICECOM/)	EDIR(LDR) (common area /EDIR/)
SPD	Average speed during the event (common area /ICECOM/)	EVBAR(LDR) (common area /EVBAR/)
TIME	Event time (common area /ICECOM/)	ETIM(LDR) (common area /ETIM/)
ESPD(ICE)	Final speed during the event (common area /ESPD/)	ESPD(LDR)
IFPC	Fire position code (common area /ICECOM/)	LFPC(LDR) (common area /LFPC/)
IDPC	Desired position code (common area /ICECOM/)	LDPC(LDR) (common area /LDPC/)
MSFP(ICE)	Fire initiation point code (common area /MSFP/)	MSFP(LDR)
LCPE(ICE)	Route array position indicator (common area /LCPE/)	LCPE(LDR)

The position of the leader from which the above offsets are measured is defined by the coordinates XLD, YLD and ZLD. These coordinates are obtained from arrays contained in common areas /ELOCX/, /ELOCY/ and /ELOCZ/, respectively. That is:

XLD = ELOCX(LDR),
YLD = ELOCY(LDR), and
ZLD = ELOCZ(LDR).

Finally, the battlefield position of ICE at the end of its event is

$$X = XLD + DELX \cdot \cos(BETA) - DELY \cdot \sin(BETA)$$
$$Y = YLD + DELX \cdot \sin(BETA) + DELY \cdot \cos(BETA)$$
$$Z = ZLD + DELZ$$

where BETA = DIR - $\pi/2$ and DIR is as recorded in Table 7.2.

Now, X and Y as defined above can be recorded directly in common area /ICECOM/ as the position coordinates of ICE. However, the Z location of ICE is a unique variable for helicopters and is recorded in common area /ELOCZ/. Thus, ELOCZ(IHCE) = Z where IHCE is the helicopter number of element ICE.

This completes our discussion of the helicopter movement model. The processing sequence of subroutine HELMOV is outlined below.

Processing Sequence

1. If ICE is a section leader, go to step 5. Otherwise, continue.

2. Determine the formation offsets for ICE.

3. Determine and record for ICE: direction of heading; speed; event time; final speed; fire position code; desired position code; route arrays position indicator; and fire initiation position indicator. Use data recorded for the section leader.

4. Determine and record the position coordinates of ICE and go to step 12.

5. If ICE is not at a retirement position, go to step 7. Otherwise, continue.

6. Set the desired position code and the event time and go to step 12.

7. Determine the event time and other controls to be used in subroutine APFDYS.

8. Use subroutine APFDYS to determine and record all the data mentioned in step 3 with the exception of the fire position code and the fire initiation position code.

10. Determine and record the fire position code and the fire initiation position code.

11. Determine and record the fuel remaining in the section.

12. The processing is complete.

VARIABLE DEFINITION INDEX

REFERENCES

1. Rheinfrank, J. J. III, and C. B. McCartney, "Aerial Platform Flight Dynamics Simulator (APFDYS)," Chapter 7 (in) Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, edited by G. M. Clark, et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

2. Parry, S. and J. J. Rheinfrank III, "Air Defense Firing and Lethality Models," Chapter 6 (in) Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, edited by G. M. Clark, et al., RF 2978 FR 71-2A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, March 1971, AD 887265L.

CHAPTER 8

HELICOPTER ELEMENT FIRING MODEL

by

G. Petty

Introduction

In this chapter the model developed to represent the firing activities of TAPCOM II helicopter elements is presented. The model represents the firing of each individual weapon which may be employed in aerial fire, and performs the resultant lethality assessment. This model interacts strongly with the helicopter fire controller model, and hence some discussion of that model and the interrelationships is included.

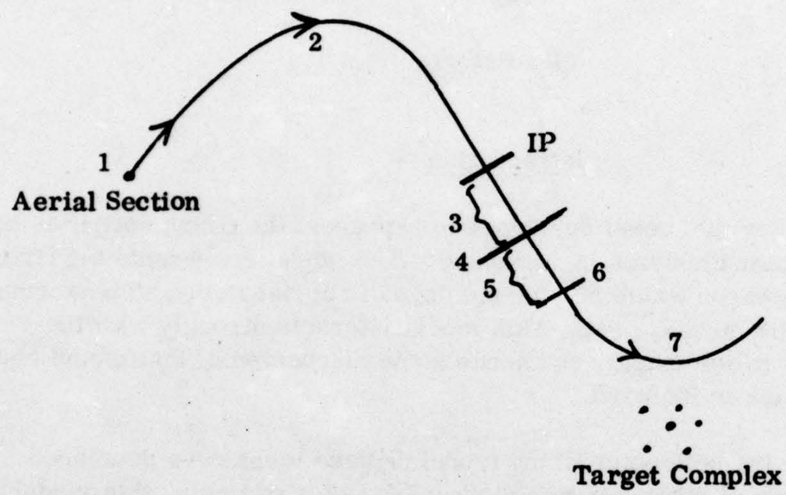
In general, the helicopter firing model utilizes techniques developed for the firing models of other non-aerial weapons. For example, this model assimilates firing data and weapon firing sequences in the same manner that the already existing armor and missile firing models are employed.

The Fire Controller

For a better understanding of the firing model a discussion and review of the weapon assignment procedures discussed in Chapter 6 is in order to present the relationship of a firing mission to individual element firing. This review will serve to define some of the concepts that are employed in the firing model and will clarify some of the requirements placed on the model by the target selection and mission assignment procedures reported in Chapter 6.

Firing missions are assigned to aerial vehicles as a section operation. The overall design for a helicopter attack involves movement by the section to a point where main weapon firing may be executed by some or all of the members of the section and subsequent movement away from this firing point. During movement to and from the main weapon firing point, suppressive fire may be employed by the vehicles to hinder attempts for return fire. Figure 8.1 shows how such a mission is sequenced.

Weapon assignments are made to individual elements in the section based on the type of weapons available, and the types of elements in the target



	Point	Action
	1.	Mission assignment.
	2.	Movement to the initial point (IP) of the attack.
Phase I	IP	Initial firing point.
Phase I	3.	Prior suppressive fire.
Phase I	4.	Main weapon firing (point fire).
Phase II	5.	After suppressive fire.
	6.	Cessation of firing.
	7.	Withdrawal from the attack point.

Figure 8.1. --Helicopter Firing Mission Sequence

complex as discussed in Chapter 6. A given helicopter may be assigned to fire suppressive fire, a main weapon fire, or both, hence two types of weapons may be employed by each vehicle.

The first type of weapon consists of rapid fire weapons which are used for suppressive fire. These weapons fire a number of projectiles in one burst, and several bursts may be fired during one firing mission by each weapon. Types of rapid fire weapons are distinguished by input data as discussed in the next section. During one firing mission an aerial vehicle may employ two different types of rapid fire weapons as will be shown.

The second type of weapon is a main weapon, or point fire weapon. This weapon may consist of a direct-fire MISTIC, indirect-fire MISTIC beam-rider missile system, or it may be another type of weapon for which the tank main gun firing model is adequate for lethality assessment. Only one round of a point fire weapon may be fired by any one element in a helicopter section during an attack.

An attack mission for a section is divided into two phases as illustrated by Figure 8.1. Phase one of the attack consists of prior suppressive fire and all point fire executions by the section. Phase two consists of all suppressive fire after the point fire.

During each phase an element may be called on to fire a maximum of two weapons, either two suppressive fire weapons or a suppressive fire weapon and a point fire weapon.

It is possible for a vehicle in the section not to receive a firing assignment because the vehicle may have insufficient ammunition of the type required for the particular target complex. Should this happen, the vehicle continues to fly with the section during the attack, however.

Table 8.1 illustrates the possible weapon firing cases which the model will accomodate for individual vehicles. This table relates the weapon combinations, the firing weapons, and the first and second phases of the firing mission. If multiple weapons are fired, each may be assigned to a different target, or they may all fire at the same element. If there is no point fire, there may be as many different targets as there are suppressive fire weapons assigned. If the vehicle is to fire a point fire weapon, any assigned prior suppressive fire will be directed at the same target. Suppressive fire following point fire may be directed at any target element including the point fire target if there was point fire by the vehicle.

The fire controller model attempts to spread the available fire of an aerial vehicle section over the target complex as completely as possible. However, targets in the complex which are more lethal to helicopters are assigned

a higher priority than the lesser lethal targets. Therefore, it is possible that not all elements in the target complex may be assigned as targets. It is also possible that not all vehicles within a section may be assigned the same type of weapon firing sequence, although coordination of the attack is maintained through the division of the attack into the two phases.

Table 8.1

Firing Assignments

Attack Phase \ Case	1	2	3	4	5	6	Weapon
1	A		A		A	A	Suppressive Fire Weapon 1
2					A	A	
1						B	Suppressive Fire Weapon 2
2	B	B				B	
1	A	A	A	A			Point Fire Weapon
Fire Type	Point Fire				No Point Fire		

Where A and B are target element numbers.

Note that A may equal B.

(blank) = no assignment.

Because of the variety of choices the fire controller model employs to develop an attack scenario, some elements in the section may be assigned only suppressive fire, while others employ only point fire. Some elements may employ a combination of point and suppressive fire.

Table 8.1 also illustrates which target is considered to be the primary, or highest priority, target during a phase when more than one weapon is assigned during the phase. Target assignments for an element may be changed between phases in the event that the target becomes a casuality or a new threat situation is perceived. It is also possible that a point fire weapon will not be fired at the end of phase one because the target element becomes a casualty during prior suppressive fire. With these possible deviations, one of the assignment sequences in Table 8.1 will be exected by an element in the section which is assigned targets during a firing event.

The Firing Model

The aerial vehicle firing model's primary responsibility is to determine which weapon and target assignments have been made to an aerial vehicle by the fire controller and to sequence the firing and lethality assessment accordingly. The present section discribes the overall firing model including the preparation of the weapons for firing and the sequencing of the firing. The next two sections describe the details of the firing of main weapons and suppressive fire weapons. The final section presents the computational procedure used by the firing model.

Subroutine HFIRE is the main program for the aerial vehicle firing model. Through this subroutine all firing is accomplished and control is passed to the armor or missile firing models as is necessitated by the particular firing assignment being processed. This program is basically divided into two sections. The first section is responsible for determining the weapons which have been assigned and for accumulating firing data necessary to represent the firing of each of these weapons. The second section is responsible for representing the firing of these weapons and for performing any resultant lethality assessment.

The sequence in which the weapons are to fire is specified by data prepared by the fire controller and is directly dependent upon which phase of the attack is presently being processed, and upon whether a point fire weapon is to be, or was, fired in the first phase. The method used to determine whether or not a particular weapon is to fire is explained in a subsequent discussion. Here we will only discuss the various possibilities that exist, given that the phase of the attack is known. The common area MSFP, subscripted by the

current element number, is used to determine which phase of the attack the firing model is processing as follows:

$$\mathrm{MSFP(ICE)} = \begin{cases} 0 & \text{first phase} \\ 1 & \text{second phase.} \end{cases}$$

Each phase of the attack is represented by one current event for an aerial vehicle and any assignments will remain constant throughout the event. For the first phase, if a point fire weapon is to be fired, the following firing possibilities are present:

a. prior suppressive fire by one weapon followed by a point fire

b. no prior suppressive fire, only point fire.

If no point fire is to be made by the vehicle, the following possibilities are present:

a. continuous suppressive fire by one weapon

b. continuous suppressive fire by two weapons.

Hence, by the knowledge that this is the first phase of the attack and if a point fire is to be made, the firing sequence can be determined and the firing data for the weapons can be prepared.

For the second phase of the attack, it is important to know if point fire was conducted in the first phase to determine the sequence. If a point fire weapon was fired, the following possibilities are present:

a. no firing

b. suppressive fire by one weapon.

If no point fire was executed, the possibilities are:

a. continuous suppressive fire by one weapon

b. continuous suppressive fire by two weapons.

Again, all firing sequences are established for the entire event.

This sequencing is important to both sections of the firing model. The first section requires knowledge of the sequencing to determine which type of

firing is to be done. The second section of the model follows this sequence in representing the fire of the weapons in the proper order.

Determining which weapons are assigned to fire, the ammunition to be fired, and the targets for the weapons is accomplished by examining common areas IHTARG and IHAMO. These common areas are filled by the fire controller model and are subscripted according to helicopter number and a combination of weapon and phase of the attack as follows (see Table 8.2):

IHTARG(I,J)

where J is the aerial vehicle number

$$I = \begin{cases} 1 \text{ suppressive fire weapon 1 - first phase} \\ 2 \text{ suppressive fire weapon 1 - second phase} \\ 3 \text{ suppressive fire weapon 2 - first phase} \\ 4 \text{ suppressive fire weapon 2 - second phase , and} \\ 5 \text{ point fire weapon - first phase only.} \end{cases}$$

Note that the numbering of suppressive fire weapons is done only to distinguish between the two weapons. This is for the user's and the model's convenience. The numbering does not necessarily indicate the importance of the weapons.

The model examines IHTARG and obtains either a target number or a zero if a particular weapon is not to fire during a phase. If a target number is obtained for the weapon, the corresponding location of IHAMO contains the ammunition code of the ammunition which is to be fired.

The target numbers are also contained in common area MDFAF and the variable LSTARG in common area ICECOM. The primary target number is contained in MDFAF. This is the target assigned to the point fire and to the after suppressive fire weapons, if this type of fire is to occur, or the primary suppressive fire weapon if no point fire is to occur. The secondary target number is contained in LSTARG. This is the target for any prior suppressive fire or for the second suppressive fire weapon if no point fire is assigned.

Once the weapons to fire have been recognized, data retrieval for the firing is performed. Data is obtained for each target to be fired upon. The target location is obtained and stored as well as the weapon code and weapon system code of the target. The target cover fraction is then computed using an aerial vehicle model subroutine HTCOV, and stored. The firer-ammu-

Table 8.2

IHTARG and IHAMO arrays
(see Table 8.1)

IHTARG(I, L) - contains target numbers.

I	Possible Weapon Assignment Cases					
1	T	0	T	0	T	T
2	0	0	0	0	T	T
3	0	0	0	0	0	T
4	T	T	0	0	0	T
5	T	T	0	T	T	0

IHAMO(I, L) - contains ammunition codes.

I	Possible Weapon Assignment Cases					
1	C	0	C	0	C	C
2	0	0	0	0	C	C
3	0	0	0	0	0	C
4	C	C	0	0	0	C
5	C	C	C	C	0	0

where

T = target number
C = ammunition code
L = aerial vehicle number.

I =
1 - Suppressive fire weapon 1 - first phase
2 - Suppressive fire weapon 1 - second phase
3 - Suppressive fire weapon 2 - first phase
4 - Suppressive fire weapon 2 - second phase
5 - Point fire weapon - first phase only.

nition characteristic is retrieved from common area IHTPRB and the firer ammunition target characteristic is retrieved from common area LTHTNK as is done in the armor firing model. Once these data values have been obtained for each weapon which is to fire, firing may commence.

The second section of the model is then entered with all information necessary to execute the firing in the proper sequence for each weapon which has been indicated. All types of firing are explained in detail in the next two sections. In general, firing is executed by determining appropriate hit probabilities, determining if a hit occurred, and if it did, determining the kill type inflicted from computed kill probabilities. All decisions as to hit and type of kill are arrived at through Monte Carlo procedures. For missile firing these proceedures are accomplished within the appropriate missile models. For the other types of firing, the decisions are made using armor firing model routine.

Although assignment of a fire mission is made to aerial vehicles with a particular sequencing of fire by the firing weapons, such as point fire preceeded by prior suppressive fire, the simulation of the firing does not necessarily follow in the same sequence. This is due to the fact that the results from firing two different weapons at two different targets during an event are completely independent. For example, should two suppressive fire weapons be assigned different targets for simultaneous firing by the fire controller, the firing model executes the complete firing event for the first weapon before firing the second. The aspect of simultaneous fire is present in that both fires are executed within the same event and using the same location of the vehicle for the initialization of fire. Firing is accomplished in its proper order when two weapons have the same target for an event. This becomes especially important in the case of two simultaneous rapid fires firing at the same target. In this case, the model executes the fires in a simultaneous manner. That is, each weapon is fired a burst at a time, with the weapon firing the next burst chosen according to the range to the target for each burst. This concept is explained more elaborately in the next section.

Rapid Fire Weapons

Rapid fire weapons fired by aerial elements during an attack consist of weapons which fire one or more projectiles per burst and which may be fired for one or more bursts during the attack. Their primary purpose in the model is as a means of suppressing enemy fire on the aerial section while preparations are made for main weapon firing and while withdrawal from the attack area is in progress. The number of bursts fired by such a weapon depends on the type of rapid fire weapon and the time allotted for suppressive fire.

This number of bursts is determined from common area BRAIR plus the time allocated to firing which is retrieved from the elapsed event time from common area ICECOM. The formula for computing the number of bursts of ammunition code I for a weapon mounted on an aerial vehicle with weapon code K is:

$$N = \frac{(TIM - BRAIR(I,4,K))}{(BRAIR(I,4,K) + BRAIR(I,1,K)/BRAIR(I,3,K)}$$

where TIM = the allowed firing time
N = the number of bursts to fire.

Common area BRAIR is dependent upon the ammunition code and the aerial vehicle weapon code, and is arranged as follows:

BRAIR(I,J,K) = rounds per burst, J = 1
desired bursts per attack, J = 2
rounds per second, J = 3
seconds between bursts, J = 4

where I = 1,...,6 - ammunition code
K = aerial vehicle weapon code.

To specify the firing characteristics for a particular weapon, common areas ICHAR, AMMOCH, FIRKON, and BRAIR are used. Common area IAMMO indicates a weapon characteristic number, dependent on firer's weapon and ammunition codes. This value is obtained by using common area LTHTNK. The ICHAR value is used as a pointer in the model into the AMMOCH and FIRKON arrays and into hit probability data arrays which will be discussed later. The muzzle velocity of the projectile is obtained from AMMOCH. The time required to fire a burst, the time for firing each round in a burst, and horizontal and vertical round to round dispersions are contained in common area FIRKON. The common area BRAIR is dependent on the firer weapon code and the ammunition code of the weapon. The number of rounds per burst, the desired number of bursts per attack, the number of rounds per second, and the minimum time between bursts are contained in this common area. Although BRAIR and FIRKON contain different information, it may be noted that some values of each can be mathmatically manipulated to give items in the other.

The time alloted to suppressive fire is predetermined by the aerial fire controller and movement models and is dependent on the phase of the attack. When the firing model receives control, the aerial vehicle has completed any movement which it is to perform during the event. The time to fire suppressive fire is determined from the time required to complete the movement

(the event time). The ranges at which the bursts are fired are computed by determining the amount of distance moved between each burst and by incrementing from the starting point of the movement event.

For the first phase of the attack, the firing time is the time required to move from the initial point of the attack to the main weapon firing point (see Figure 8.1) minus the time required to prepare the main weapon for firing. This preparation time is discussed in the next section. The firing time for the second phase is the time from the firing of the main weapon to the conclusion of the movement for the attack. Hence, from the firing time and the weapon characteristics, the number of bursts a weapon is to fire in a phase can be calculated before each firing action.

After each rapid fire burst is executed, a lethality assessment on the target is performed. To accomplish this, hit probabilities for the weapon must be determined. Two classes of hit probabilities are incorporated into the model. The first type is the probability prior to obtaining a hit on the target; i.e., repititious first round cases, and the second type is the probability used subsequent to a hit.

The first round hit probabilities are determined by a new subroutine HHPROB. This probability is calculated by using horizontal and vertical fixed biases and first round dispersions contained in the existing common areas RPSIGX, RPSIGY, TSDXM and TSDYM and the dispersions from common area FIRKON. The fixed bias and first round dispersion arrays are dependent on the ICHAR value and the range to the target. When the total deviation in the aim point has been determined, a previously existing subroutine HIPROB calculates the actual hit probability.

The hit probability for cases in which a hit has already occured is determined by existing subroutine HPROB using only the dispersion values from common area FIRKON and subroutine HIPROB.

After the hit probability for a burst has been computed, Monte Carlo proceedures are used to determine if a hit and kill occurred during the burst. If no hit occurred, there is no lethality assessment, and the next round is processed. If an initial hit occurred, the appropriate first round flag, LFRND for primary targets, KFRND for secondary targets, is turned off. These first round flags are set to one inthe fire controller model. After a hit occurs they are set to zero.

Subroutine NEUTIM is called to determine a neutralization time for the target each time a hit occurs. The determination of a hit and kill is performed by subroutine TLETH dependent on the number of rounds fired within the burst. Subroutines TLETH, HPROB, and HIPROB are part of the armor module and

are described in detail in the presentation of that model (see references 1 and 2). Should a kill occur, subroutine CASELE performs the casualty processing. If the kill includes both mobility and firepower (MF), or it is a total kill (K), the target is deleted and the weapon is inactive for the remainder of the engagement. The computational proceedure for rapid fire execution is as follows:

1. Determine the aerial vehicle number of the helicopter; i.e., set L = LHICE(ICE).

2. Choose the ammunition type for the weapon, IAMO, from IHAMO(J, L) where J is dependent on attack phase and rapid fire weapon.

3. Determine the firing time, TIM, from the elapsed movement time.

4. Determine the aerial vehicle weapon code, LC, and the aerial unit weapon code, LWC.

5. Determine the number of bursts to fire; i.e., set

$$N = \frac{(TIM-BRAIR(IAMO, 4, LWC))}{(BRAIR(IAMO, 4, LWC)+BRAIR(IAMO, 1, LWC)/BRAIR(IAMO, 3, LWC))}.$$

6. Determine the target number, ITRG from IHTARG(J, L).

7. Determine the target weapon code, ITC, and weapon system code, ITS.

8. Determine the aerial vehicle position at the initiation of fire, (X, Y, Z).

9. Determine the target position (XT, YT, ZT).

10. Determine the initial range, RANGE.

11. Compute target cover fraction, COV by calling subroutine HTCOV.

12. Determine the firer-weapon characteristic; i.e., set IC = IHTPRB(IAMO, LC).

13. Determine the firer-weapon-target lethality characteristic; i.e., set IK = LTHTNK(IAMO, LC, ITC).

14. Determine the position of the aerial vehicle at the end of firing (XH,YH,ZH).

15. Determine the distance moved between each burst in each plane; i.e., set

$$XM = (XH-X)/N$$
$$YM = (YH-Y)/N$$
$$ZM = (ZH-Z)/N$$

16. Set IB = 1 for firing each burst.

17. If the target is MF or K killed, go to step 33.

18. Determine range at the burst point; i.e., set

$$XR = XT-XH+(I-1)*XM \text{(same for YR,ZR)}$$

$$RANGE = \sqrt{XR^2+YR^2+ZR^2}$$

19. Determine speed and aspect factors by calling subroutine SPDASP.

20. Check LFRND or KFRND for a first round case, and if so, go to step 23.

21. Determine non first round case hit probability HITPR by calling subroutine HPROB.

22. Go to step 24.

23. Determine first round hit probability by calling subroutine HHPROB.

24. Call TLETH to determine result of fire, IHIT and IKILL.

25. If there was no hit; i.e., if IHIT = 0, go to step 31.

26. If this was not a first round case, go to step 28.

27. Set the proper first round flag, LFRND or KFRND.

28. Call subroutine NEUTIM to assess a neutralization time.

29. If there was no kill; i.e., if IKILL = 0, go to step 31.

30. Call subroutine CASELE to record the casualty.

31. Increment the burst; i.e., IB = IB+1.

32. If there are remaining bursts; i.e., if IB≤N, go to step 17.

33. Rapid fire for this weapon is concluded.

It may be noted here that this sequence above is that followed by each rapid fire event. Hence, within the computational proceedure for the entire firing model at the end of the present chapter does not contain the numerous duplicate copies of this proceedure which would be required. Instead a reference is given to the above proceedure. A seperate subroutine was not created for this proceedure because each weapon obtains firing data from different sources, and because the proceedure is interspersed with various sections of logic for simultaneous rapid fire.

An unusual circumstance occurs when there are two suppressive fire weapons firing simultaneously at the same target. These weapons are fired simultaneously by the model in the following manner.

1. Compute range for each weapon's first burst.

2. Fire a burst by the weapon with the greatest range.

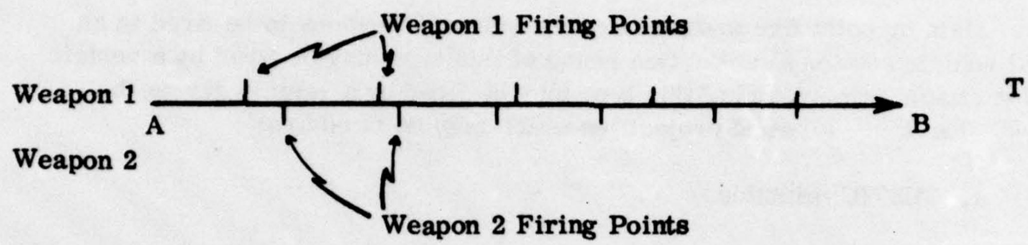
3. If the target is killed, conclude firing by both weapons.

4. Compute the next range for the weapon which just fired. (If the burst count has expired for the weapon, set the range to zero).

5. If both weapons have zero ranges, conclude firing.

6. Go to step 2.

The method for computing the number of bursts to fire, and assessing lethality remain unchanged from the proceedure described previously for rapid fire. These sequencing steps are merely interspersed with the normal rapid fire execution. This method is included because the variety of rapid fire weapons which may be employed may require a different number of bursts to be fired by each weapon. This would result in staggered burst firing. (See Figure 8.2).



$\overrightarrow{AB}$ = vehicle's approach to target.

T = target element.

Firing is accomplished from A to B in the following manner:

BURST	WEAPON
1	1
2	2
3	1
4	2
5	2
6	1
7	2
8	1
9	2
10	1
11	2

Figure 8.2. --Simultaneous firing by two rapid fire weapons at the same target.

Main Weapons

Main or point fire weapons are the primary weapons to be fired in an aerial vehicle section attack. One round of this type may be fired by a vehicle section attack. One round of this type may be fired by a vehicle during the attack. The three types of projectiles which may be fired are:

a. MISTIC missiles

b. Beam rider missiles

c. A weapon which may be simulated by the armor tank main gun firing model.

All three of these types of weapons existed in DYNCOM prior to the introduction of the aerial vehicle model, and are therefore duscussed in detail in previous reports describing DYNCOM The MISTIC missile model is discussec in reference 3 and in Chapters 2 and 3, the beam rider missile model in reference 4, and the armor firing model in references 1 and 2. The aerial vehicle firing model is responsible for determining which point fire weapon has been chosen by the aerial fire controller, assessing a firing or preparation time, dependent on the weapon fired, and dispatching control to the appropriate firing model.

The type of point-fire weapon to be fired is determined by using the ammunition code of the assigned weapon (IAMMO) and the array IHDFMC as follows:

1. If IAMMO = IHDFMC(1, LWC), helicopter L is to fire a MISTIC missile.

2. If IAMMO = IHDFMC(2,LWC), helicopter L is to fire a beam rider missile.

3. Otherwise, helicopter L is to fire a weapon that can be represented by the armor module main gun firing model.

Should the target already be a MF or K kill, the mission is not initiated.

When a MISTIC missile is to be fired, no firing time is assessed within subroutine HFIRE; instead, a preparation time is computed within the aerial

fire control model then, within subroutine HFIRE, subroutine HLNCH is called to create the missile. The missile becomes an element at the time of launch and from then on the flight of the missile as an element is handled by the MISTIC flight model (Chapter 2 and reference 3). Hence, the aerial firing model relinquishes all control after firing and does not assess lethality of MISTIC missiles.

For a beam rider missile, the flight time is computed within the beam rider missile model. Subroutine SHILLY is called to simulate the flight to its conclusion within the present event and returns to the aerial firing model with flags set for indicating the hit and kill results of the firing. If there was a hit, subroutine NEUTIM is called to determine the neutralization time for the target. Should a kill result, subroutine CASELE is called to record the casualty .

For a weapon that is to be represented by the armor firing model, subroutine HFTIME is called to determine firing time and also if a misfire occurred. If there was no misfire, subroutine FIRMOD is called to execute the firing and lethality assessment. In this case as in the firing of a beam rider missile, subroutines NEUTIM and CASELE are called if there was a hit.

The computational proceedure for main weapon firing is as follows:

1. Determine the aerial vehicle number of the helicopter; i.e., L = LHICE(ICE).

2. Determine the target number; i.e., set ITARG = IHTARG(5, L).

3. Determine aerial vehicle position at the time of firing, (XH; YH, ZH).

4. Determine the target position at the time of firing, (XT, YT, ZT).

5. Set the range, RANGE, of the fire.

6. Determine the target weapon code, ITC, and weapon system code, ITS.

7. Determine the aerial vehicle unit weapon code, LC.

8. Determine the ammunition characteristic of the fire; i.e., set IAMMO = IHAMO(5, L).

9. Determine the firer ammunition characteristic ; i.e., set ICHAR = IHTPRB(IAMMO, LC).

10. Determine the fires-ammo-target lethality characteristic; i.e., set KCHAR = LTHTNK(IAMMO, LC, ITC).

11. Determine the target cover fraction COV by calling subroutine HTCOV.

12. If the target is MF or K killed, go to step 29.

13. If the point fire is a beam rider missile; i.e., if IAMMO = IHDFMC(2, LWC), go to step 24.

14. If the point fire is a MISTIC; i.e., if IAMMO = IHDFMC(1, LWC) go to step 27.

15. Call subroutine HFTIME for the flight and firing time FTIM, and the misfire flag, MFIRE.

16. Set the flight time; i.e., TFLY(L) = FTIM.

17. If there was a misfire; i.e., if MFIRE = 1, go to step 29.

18. Fire the point fire with the tank main gun logic by calling subroutine FIRMOD with hit, IHIT, and kill, IKILL, flags.

19. If there was no hit; i.e., if IHIT = 0, go to step 29.

20. Call subroutine NEUTIM to assess a neutralization time.

21. If there was no kill; i.e., if IKILL = 0, go to step 29.

22. Call subroutine CASELE to record the casualty.

23. Go to step 29.

24. Call subroutine SHILLY to fire the beam-rider missile with hit, IHIT, and kill, KILL, flags.

25. Set the actual flight time of the missile from common TALK; i.e., TFLY(L) = TALK(1).

26. Go to step 19.

27. Call subroutine HLNCH to create a MISTIC missile element.

28. Set the flight time to zero; i.e., TFLY(L) = 0.

29. Point fire is concluded.

In the computational proceedure for the firing model at the end of this chapter, the above proceedure is not repeated, but is merely referenced.

Auxillary Subroutines

Four new routines are included in the aerial vehicle firing model which perform single tasks. Two of the routines HFTIME and HLNCH are related to point fire. The third, HTCOV, is included as a subroutine because of the numerous requirements for the contained logic. The fourth, HHPROB, is concerned with first round hit probabilities for rapid fire weapons.

Subroutine HFTIME is a modification of subroutine FTIME for the armor firing model. It's purpose is to assess a firing time for a point fire weapon which is to be simulated by the armor firing model, and to determine probabalistically if the weapon misfired. The flight time for the projectile is dependent on the range and muzzle velocity and is computed by the formula:

FTIM = RANGE/AMMOCH(1,ICHAR)

where ICHAR is the firer-weapon characteristic from common IHTPRB. A misfire is determined by a Monte Carlo proceedure using common area PMISF. If a misfire should occur, the flight time is set to zero, and no further processing of the point fire is done.

Subroutine HHPROB is also a modification of logic which appears in subroutine FTIME. It calculates the first round hit probabilities for aerial vehicle rapid fire events.

Horizontal and vertical round to round dispersions, RTRX and RTRY, are obtained from common area FIRKON. First round horizontal and vertical disperssions, SDXT and SDYT, are obtained from common areas TSDXM and TSDYM. These common areas are dependent upon the range intervals from common area HPRNG, and the rapid fire weapon characteristic. Next, the horizontal and vertical fixed biases, SDXM and SDYM, are obtained from common areas RPSIGX and RPSIGY. These common areas are also dependent upon the range and the weapon characteristic. The dispersions and biases are then accumulated by:

$$SDXT = \sqrt{RTRX^2 + SDXT^2 + SDXM^2}$$

$$SDYT = \sqrt{RTRY^2 + SDYT^2 + SDYM^2}$$

and subroutine HIPROB is called to determine the actual hit probability.

Subroutine HTCOV computes the target cover fraction for each target an aerial vehicle may engage. The cover fraction is computed from the midpoint of the trajectory in the following manner:

1. Determine the range to the target, RANGE.

2. Determine the target weapon code, ITC.

3. Compute the target height, TGTH, from common area TGTDIM.

4. Set XI, YI, ZI to the aerial vehicle position.

5. Set XT, YT, ZT to the target element position.

6. Find the midpoint of the trajectory; i.e., set

$$XV = (XI+XT)/2$$
$$YV = (YI+YT)/2$$
$$ZV = (ZI+ZT)/2+1.22*\left(\frac{RANGE}{AMMOCH(1, ICHAR)}\right)^2$$

7. Compute the elevation of the line of sight, HV, by calling subroutine ELVATE.

8. Find the difference between the height of the midpoint of trajectory and the line of sight; i.e., set HV = ZV-HV.

9. Call subroutine SLOSS to determine the cover fraction SIGHT.

10. Compute the amount of the target visible; i.e., set

$$HTCOV = SIGHT*TGTH-|TGTDIM(7,ITC)|.$$

11. Return to HFIRE.

Subroutine HLNCH is called when the point fire weapon is a MISTIC missile. This subroutine creates the missile as a battlefield element, and sets the missile clock to activate the MISTIC flight model. The following is a representation of the processing accomplished in subroutine HLNCH:

1. Determine the number of the current element, ICE.

2. Determine the launcher number, NUM, of the vehicle.

3. Determine the aerial vehicles position (XE, YE) and current clock time CLOCK.

4. Determine the weapon code, KWCOD, and aerial unit weapon code, LCOD.

5. Determine the ammunition code of the fire; i.e., set ITYP = IHDFMC(1, LCOD).

6. Decrement the ammunition supply by calling subroutine AMMOD.

7. Set SDXLI, SDYLI, SDXLM, and SDYLM to zero.

8. Monte Carlo for a failure, by using common zrea PHNG: if there was no failure, go to step 9; otherwise, go to step 18.

9. Determine the missile launch time; i.e., set TM = CLOCK + TIM - R, where TIM is the elapsed event time and R is a random number from a uniform distribution defined on the interval (0, 1).

10. Create a missile element by calling subroutine CREATM.

11. Record firing data; i.e., launch time, helicopter coordinates, target number, etc., in common areas LNSET and OPEN.

12. If the aerial unit is performing a search and destroy mission, go to step 15.

13. Reset FO numbers and increment FO availability time by setting
    NF = NUMART + NUM
    IFO = KFO(NF)
    LFO = NOBVH(IFO) and
    TIFRDY(IFO) = TM.

14. Go to step 16.

15. Increment launcher availability time; i.e., set TDFRDY(NUM) = TM.

16. Record missile launch; i.e., set LFLAG(NUM) = 2.

17. Set up missile flight data by calling subroutines LNBFLT and MICONP.

18. Return to HFIRE.

Computational Proceedure

This section contains the overall computational proceedure for the aerial vehicle firing model. This proceedure is simplified because much of the contained logic is presented in proceedure form in previous sections. Hence, when the reader observes statements such as 'Retrieve data for rapid fire weapon one,' or 'Fire the point fire weapon,' the proceedures in the appropriate sections should be referenced. The main purpose of this proceedure is to illustrate how the model coordinates the weapon firing sequences.

1. Determine aerial vehicle number of current element; i.e., set
   L = LHICE(ICE).

2. Determine aerial vehicle's current position, (X, Y, Z).

3. Determine aerial vehicles position at the beginning of the event.

4. Set fire flags for each weapon; i.e.,
   point fire - FIRE 1 = .FALSE.
   rapid fire - FIRE 2 = .FALSE.
   rapid fire - FIRE 3 = .FALSE.

5. If this is the second phase of the mission; i.e., if
   MSFP(ICE) = 1,
   go to step 20.

6. If there is to no point fire; i.e., if IHTARG(5, L) = 0,
   go to step 13.

7. Set FIRE 1 = .TRUE.

8. Obtain point fire weapon data.

9. If there is no prior suppressive fire; i.e., if
   IHTARG(1, L) = 0,
   go to step 31.

10. Set FIRE 2 = .TRUE.

11. Obtain firing data for rapid fire weapon 1.

12. Go to step 31.

13. If there is no primary suppressive fire (rapid fire weapon 1) ; i.e., if IHTARG(1,L) = 0, go to step 45.

14. Set FIRE 2 = .TRUE.

15. Obtain firing data for rapid fire weapon 1.

16. If there is no secondary suppressive fire (rapid fire weapon 2); i.e., if IHTARG(3,L) = 0, go to step 31.

17. Set FIRE 3 = .TRUE.

18. Obtain firing data for rapid fire weapon 2.

19. Go to step 31.

20. If there was not point fire in the first phase; i.e., if
    IHTARG(5,L) = 0,
    go to step 25.

21. If there is no after suppressive fire (rapid fire weapon 2); if IHTARG(4,L) = 0, go to step 45.

22. Set FIRE 3 = .TRUE.

23. Obtain firing data for rapid fire weapon 2.

24. Go to step 31.

25. If there is no primary suppressive fire (rapid fire weapon 1); i.e., if IHTARG(2,L) = 0, go to step 45.

26. Set FIRE 2 = .TRUE.

27. Obtain firing data for rapid fire weapon 1.

28. If there is no secondary suppressive fire (rapid fire weapon 2); i.e., if IHTARG(4,L) = 0, go to step 31.

29. Set FIRE 3 = .TRUE.

30. Obtain firing data for rapid fire weapon 2.

Firing Begins

31. If there is no point fire; i.e., if FIRE 1 = .FALSE., go to step 37.

32. If there is no prior fire; i.e., if FIRE 2 = .FALSE., go to step 35.

33. Fire rapid fire weapon 1.

34. If the target is a casualty; i.e., if LKILL (ITARG) ≥ 3, go to step 45.

35. Fire the point fire weapon.

36. Go to step 45.

37. If there is simultaneous rapid fire; i.e., if FIRE 2 = .TRUE., and FIRE 3 = .TRUE., go to step 43.

38. If rapid fire weapon 1 is not to fire; i.e., if FIRE 2 = .FALSE., go to step 40.

39. Fire rapid fire weapon 1.

40. If rapid fire weapon 2 is not to fire; i.e., if FIRE 3 = .FALSE., go to step 45.

41. Fire rapid fire weapon 2.

42. Go to step 45.

43. If the rapid fire weapons do not have the same target; i.e., if
    ITARG 2 ≠ ITARG 3,
    go to step 38.

44. Execute simultaneous fire at the same target by both rapid fire weapons. If at any time the target is killed, go to step 45.

45. The aerial vehicle firing is complete.

VARIABLE DEFINITION INDEX

Variable	Definition	Variable	Definition
AMMOCH	p. 8 -10 (input)	MFIRE	p. 8 -18
BRAIR	p. 8 -10 (input)	MSFP	p. 8 -5
CLOCK	p. 8 -21	N	p. 8 -10
COV	p. 8 -12	NUM	p. 8 -21
FIRE1	p. 8 -22	PHNG	p. 8 -21 (input)
FIRE2	p. 8 -22	PMISF	p. 8 -19 (input)
FIRE3	p. 8 -22	RANGE	p. 8 -12
FIRKON	p. 8 -10 (input)	RPSIGX	p. 8 -11 (input)
FTIM	p. 8 -18	RPSIGY	p. 8 -11 (input)
HPRNG	p. 8 -19 (input)	RTRX	p. 8 -19 (input)
HV	p. 8 -20	RTRY	p. 8 -19 (input)
IAMMO	p. 8 -10	SDXM	p. 8 -19 (input)
IAMO	p. 8 -12	SDYM	p. 8 -19 (input)
IB	p. 8 -13	SDXT	p. 8 -19 (input)
IC	p. 8 -12	SDYT	p. 8 -19 (input)
ICHAR	p. 8 -10	SIGHT	p. 8 -20
IHAMO	p. 8 -8	TALK	p. 8 -18
IHDFMC	p. 8 -16 (input)	TDFRDY	p. 8 -21
IHIT	p. 8 -13	TFLY	p. 8 -18
IHTARG	p. 8 -8	TGTDIM	p. 8 -20 (input)
IHTPRB	p. 8 -9 (input)	TGTH	p. 8 -20
IK	p. 8 -12	TIM	p. 8 -10
IKILL	p. 8 -13	TM	p. 8 -21
ITARG	p. 8 -17	TSDXM	p. 8 -11 (input)
ITC	p. 8 -12	TSDYM	p. 8 -11 (input)
ITRG	p. 8 -12	X	p. 8 -12
ITS	p. 8 -12	XE	p. 8 -21
ITYP	p. 8 -21	XH	p. 8 -13
KCHAR	p. 8 -18	XI	p. 8 -20
KFRND	p. 8 -11	XM	p. 8 -13
LC	p. 8 -12	XR	p. 8 -13
LCOD	p. 8 -21	XT	p. 8 -12
LFLAG	p. 8 -21	XV	p. 8 -20
LFRND	p. 8 -11	Y	p. 8 -12
LHICE	p. 8 -12	YE	p. 8 -21
LNSET	p. 8 -21	YH	p. 8 -13
LSTARG	p. 8 -8	YI	p. 8 -20
LTHTNK	p. 8 -9 (input)	YM	p. 8 -13
LWCOD	p. 8 -21 (input)	YR	p. 8 -13
MDFAF	p. 8 -8	YT	p. 8 -12

Variable	Definition
YV	p. 8 -20
Z	p. 8 -12
ZH	p. 8 -13
ZI	p. 8 -20
ZM	p. 8 -13
ZR	p. 8 -13
ZT	p. 8 -12
ZV	p. 8 -20

REFERENCES

1. Wilson, N. B., "Firing Model," Chapter 8 (in) The Tank Weapon System, edited by A. Bishop and G. Clark, RF 573 AR 69-2B (U), Systems Research Group, The Ohio State University, Columbus, Ohio, September 1969, AD 864920.

2. Hutcherson, D. C. and G. Petty, "Miscellaneous Topics," Chapter 11 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, edited by G. M. Clark and D. C. Hutcherson, RF 2978 FR 71-3A (U), Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

3. Parry, S. H., "Indirect Fire," Chapter 1 (in) The Land Combat Model, Classified Annex, edited by G. Clark, et al., RF 2376 FR 71-5 (S), Systems Research Group, The Ohio State University, Columbus, Ohio, January 1971, AD 517621.

4. Clark, G. M., "Beam Rider Missile System," Chapter 8 (in) The Land Combat Model (DYNCOM), RF 2376 FR-1 (U), edited by A. B. Bishop and G. M. Clark, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

CHAPTER 9

MISCELLANEOUS TOPICS

by

D. C. Hutcherson and G. Petty

Introduction

This chapter is included for the purpose of presenting discussions of various topics that are not conveniently presented in other chapters of this volume. The reader will find that the topics discussed either involve modifications to the DYNCOM program that were necessitated by developments reported elsewhere in this report or they involve changes that have been made to provide a more representative simulation model.

The topics, in the order they appear in this chapter, are as follows:

1. Helicopter Casualty Procedures
2. Cover and Concealment Procedures
3. Fire Controller Revisions.

Helicopter Casualty Procedures

The air-defense weapons operations model developed for DYNTACS X was reported in detail in reference 1. The model is designed to represent acquisition of aerial targets by ground-based air-defense weapons, aerial target selection and assignment, ammunition selection, and the firing and flight processes of air-defense projectiles. The model is capable of ascertaining whether or not one or more hits occur on a specified target helicopter during a firing event, and if hits occur, the types of damage inflicted. However, the description of lethality prepared by the model consists of a set of generalized lethality codes that are not in themselves descriptive of the effects produced on the target. These codes must be translated in order to arrive at an actual description of the effects of damage. The discussion in reference 1 did not include a description of this translation process nor did it describe the DYNCOM processing procedures of casualty helicopter elements. Both of these topics are discussed in the paragraphs which follow.

Background and Definitions

The air defense firing and lethality assessment models are incorporated into the existing direct-fire ballistic weapon firing model programmed as subroutine FIRMOD. This subroutine is called from the DYNCOM main program when any kind of direct-fire weapon, except a MISTIC missile or beam-rider missile is to be fired from a ground-based vehicle. The subroutine has internal logic to assure that air-defense firing is treated as discussed in reference 1.

When subroutine FIRMOD is used to represent an air-defense firing event, the variable IHIT is set as follows:

$$\text{IHIT} = \begin{cases} 1, \text{if the target helicopter was hit by one or more} \\ \quad \text{projectiles during the event} \\ 0, \text{ if otherwise} \end{cases}$$

Moreover, if a hit occurred (IHIT = 1) the battle time that the hit(s) occurred is recorded as THIT. Both of these variables appear in the calling list of subroutine FIRMOD, and thus, they are available to the DYNCOM main program.

Finally, the effects produced on the target helicopter element are recorded in the lethality code array, LTHCOD, which is stored in COMMON/LTHCOD/. The array entries are defined as follows:

$$\text{LTHCOD}(I, M) = \begin{cases} 1 & \text{if damage of type M was inflicted on} \\ & \text{helicopter element I during the firing event} \\ 0 & \text{if otherwise} \end{cases}$$

where

$$\begin{aligned} I &= 1, \ldots, N, \\ M &= 1, \ldots, \text{MAXK}, \\ N &= \text{total number of helicopter elements being represented,} \\ \text{MAXK} &= \text{total number of lethality codes represented.} \end{aligned}$$

As explained below, a constraint MAXK $\leq$ 9 is imposed to insure proper processing. Also, for a particular firing event against helicopter L, we need concern ourselves only with the vector LTHCOD(L, M) M = 1, . . . , MAXK.

After subroutine FIRMOD is called in the DYNCOM main program, it may be determined whether or not the firing resulted in one or more hits by the calling parameter IHIT > 0 as explained previously. If this is the case, then the effects of the hits must be assessed. Otherwise, no further processing is required to assess lethality.

Lethality Assessment

To assess lethality, subroutine CASHEL, whose flow chart appears in Volume 2, is used. This program translates the entries in the LTHCOD array into physical descriptors of the damage inflicted on the target helicopter element L. The assumptions of this model are as follows:

1. Damage of type M, as indicated by an entry in the array LTHCOD, can be described as reducing the firepower of the helicopter element by denying the use of certain weapons with which the element was originally equipped.

2. Damage of type M, as indicated by an entry in the array LTHCOD, can be described as limiting the amount of time the helicopter element can remain over the battlefield.

To implement the above assumptions, two arrays of input data are used. First, the array KILFIR specifies the reduction of firepower resulting from damage. Second, the array, FMOKIL, specifies the time limits placed on the target element by the inflicted damage. Both arrays are contained in common areas bearing the respective array names (see Volume 2).

The array KILFIR is a two-dimensional array of integers as follows:

KILFIR(I, J) = firepower reduction inflicted by damage of type I on a helicopter with helicopter weapon code J

where I = 1,..., MAXK
J = 1,..., N
MAXK is the maximum number of lethality codes as defined previously, and
N is the number of helicopter weapon codes

The reader will recall from Chapter 6, that the same weapon code applies to each element in an aerial section and the helicopter weapon code is J = LWCOD(K) - MAXLWC for element K where LWCOD(K) is the weapon code of K and MAXLWC is the maximum vehicular weapon code exclusive of aerial vehicles as defined in COMMON/NUMBER/.

Firepower reduction as specified by an entry in the array, KILFIR, is an integer consisting of MNAMO digits defined as follows:

$$\text{KILFIR(I, J)} = \left\{ \delta K;\ K = 1, \ldots, \text{MNAMO} \right\}$$

where

$$\delta K = \begin{cases} 1, \text{ if ammunition of type K can no longer be fired as a result of damage of type I} \\ 0, \text{ if otherwise} \end{cases}$$

The variable, MNAMO, is the maximum number of ammunition types aboard each element as specified in COMMON/NUMBER/; and the subscript, K, corresponds to the subscript for ammunition type in COMMON/LAMMO/. For example, KILFIR(I, J) = 100101, would specify that damage of type I on helicopter weapon code, J, removes the capability of delivering ammunitions 1, 4 and 6. Furthermore, KILFIR(I, J) = 1001, would specify inability to deliver ammunitions of types 3 and 6, and so on. Note that more than one type of damage may deny the use of the same ammunition type.

The array, FMOKIL, is also a two dimensional array arranged in a fashion identical to KILFIR. However, an entry in FMOKIL specifies time constraints as follows:

FMOKIL(I, J) = time remaining (in seconds) to a helicopter with weapon code J after a hit that inflicts damage of type I.

The dimensions are seconds, and the size of the array remains MAXK*N.

Within subroutine, CASHEL, an accounting is first made of the different types of firepower damage that have been inflicted by constructing the number

$$\text{KAMR} = \sum_{I=1}^{\text{MAXK}} \text{KILFIR(I, J)*LTHCOD(L, I)}$$

where L is the helicopter number of the target element and J is the helicopter weapon code. The variable, KAMR, has the same properties as an entry in KILFIR, except the individual digits of the number may be greater than one

(because of duplicate ammunition damage inflicted by different lethality codes). However, we limit MAXK to a number less than ten so that an individual column in the sum always adds to a number less than ten. Thus, the maximum value of KAMR may be MNAMO digits having the value 9.

After the total effects have been assessed, we simply zero the ammunition supply of the affected ammunition types. That is, if digit M in KAMR is positive, we delete the supply of ammunition of type M aboard the target element (LAMMO(M, IT) = 0 for element IT). From then on, helicopter L will be unable to deliver fire of the type that was damaged.

Next, in CASHEL, we assess the time limits imposed on the target element by the various damage types. That is, we construct the number

$$\text{TIMR} = \min_{1 \leq I \leq \text{MAXK}} \{\text{FMOKIL}(I, J);\ \text{LTHCOD}(L, I) > 0\}$$

where L is the helicopter number and J is the weapon code as before. Now, the time that the helicopter L will have to retire as the result of the damage is TIMR + THIT, where THIT is the battle time at which the hit(s) occurred as determined in FIRMOD. This value is then compared against the retirement time already recorded for the target helicopter to determine whether the latest recorded damage places more severe constraints on the helicopter than already exist. Thus,

$$\text{TRET}(L) = \min(\text{TRET}(L), \text{TIMR}).$$

The array, TRET, is contained in COMMON/TRET/ and represents the latest retirement time recorded for each helicopter element L. It must be initialized with large values exceeding the maximum expected battle duration.

Helicopter Casualty Processing

The procedures of subroutine CASHEL described above do not remove helicopter elements from the battle when they become casualties. Subroutine CASHEL only translates the lethality codes of the air defense model into physical attributes that describe the effects of air defense weapon lethality. Therefore, we still need a procedure to remove helicopters from the battle when they become casualties. This procedure is implemented in subroutines GETHEL and HRAPUP, whose flow charts appear in Volume 2.

Subroutine GETHEL is always called from DYNCOM main program as the first procedure when the current element is a helicopter. This subroutine then determines whether or not the helicopter that is current is a casualty. If it is a casualty, processing occurs to remove the element from the battle. Otherwise, no processing is required.

Time remaining to the current helicopter L is the criterion used to determine whether or not it is a casualty. This time is

DELT = TRET(L) - ECLOCK(ICE)

where TRET(L) is the recorded retirement time for L as discussed previously, and ECLOCK(ICE) is the current clock time of the element ICE, which corresponds to L (L = LHICE(ICE)). The following conventions are used:

if DELT > TCRIT(J), element ICE is not a casualty,
if 0 < DELT ≤ TCRIT(J), element ICE is an MF kill (LKILL(ICE) = 3), and
if DELT ≤ 0, element ICE is a K kill (LKILL(ICE) = 4).

Here, J, is the helicopter weapon code as defined previously and the array, TCRIT, is input and is contained in COMMON/TCRIT/.

If LKILL(ICE) is recorded positive by the above convention, ICE must be removed from the battle. However, the steps differ depending upon whether or not ICE is the last remaining survivor in a section.

Section Contains Other Survivors

When the section contains other survivors, element ICE is simply removed from the section organization, the section organization is revised to fill the vacated position, and the event time for ICE is recorded as a large positive number. Then, if ICE was the leader of his maneuver unit, MANUN; i.e., if MANLDR(MANUN) = ICE, the element that took ICE's position in the section

organization is recorded as the new maneuver unit leader. Finally, the variable LIMOV(ICE) is set to zero to indicate that ICE did not move in its last event. This value is used by the intelligence model to disallow detection of casualty or inactive helicopter elements as explained in Chapter 5.

When processing control is returned to the DYNCOM main program from GETHEL, the positive event time recorded for ICE is a key to the fact that the element is not to be processed for the rest of its event. Therefore, the event is aborted and control is passed directly back to the sequence controller. The event counter is not incremented in this case since a valid event did not take place. The sequence controller increments the clock of the casualty element by the large event time and the element never becomes current again.

Entire Section is a Casualty

When the casualty element is the sole remaining survivor, the section has actually become a casualty. To facilitate processing, the element is not removed from its section and the event time is recorded as zero. However, ICE is recorded as a casualty as explained earlier. The convention outlined above causes the following processing to occur in the DYNCOM main program:

1. The zero event time permits the element to be processed through the regular helicopter logic of the main program.

2. Intelligence and communications for the element will be updated to account for activities of the previous event.

3. Since the element is the sole survivor of the section it will occupy position one in the section organization and will be treated as a section leader. Therefore, it will be processed through the section movement controller (subroutine HELCON).

4. The section movement controller, recognizing the element as the sole survivor (actually dead) of the section, will perform bookkeeping that is normally performed for any other section that is retiring from the battlefield. This processing will effectively remove the section from the maneuver unit organization as explained in Chapter 6.

5. The main program regular helicopter logic is then interrupted after the return from subroutine HELCON. Instead of being processed through the movement model (subroutine HELMOV), the element

is processed through subroutine HRAPUP. This program is described below and its flow charts appears in Volume 2.

6. Finally, control is passed to the sequence controller which updates the element's clock by the large event time recorded in HRAPUP. Thus, the casualty element will never become the current element again. Moreover, the section will have been permanently removed from the maneuver unit organization. Note that the event is again treated as a dummy event so that the event counter is not incremented.

Subroutine HRAPUP simply computes the large event time used for casualty elements and removes the casualty element from its section. Thus, this subroutine performs the same processing that the organization adjustment logic of subroutine GETHEL performs. However, the logic is simpler since no other elements in the section exist.

Cover and Concealment Procedures

With the incorporation of models of aerial vehicle operations into DYNCOM, it has become necessary to modify the method by which the elevation of a vehicle is determined. This modification is required to account for the possibility that a vehicle may now be elevated above the terrain. Since elevation of a target or an observer drastically affects the amount of cover and concealment a vehicle has, it is extremely important the elevations be determined accurately. The existence of a line of sight might even be altered if the elevation of the target or the observer were incorrectly computed.

Background

In the original model the absolute height of an observer could be determined by the relationship

$$ZO = HV + EM + HTER$$

where HV is the height of the observer vehicle as determined from entries in COMMON/TGTDIM/, EM is the deviation of the observer vehicle elevation from the macro-terrain profile as recorded in COMMON/EMICR/ and HTER is the elevation of the macro-terrain profile at the observer's position XO, YO. This last value was determined by function ELVATE as follows:

$$HTER = ELVATE(XO, YO).$$

If a line of sight to a target vehicle were desired, the absolute elevation of the target vehicle ZT could be determined in a manner similar to that used for ZO. If the line of sight were to a point on the ground (XT, YT), then

ZT = ELVATE (XT, YT).

Elevated Vehicles

The relationships for ZO and ZT do not apply directly if the observer or the target is a helicopter and is elevated. Instead, the relationship

ZO = HV + HTER + ELOCZ(L)

or

ZT = HV + HTER + ELOCZ(M)

apply, where L and M are the helicopter numbers of the observer and target vehicles, respectively; and ELOCZ(I) is the recorded elevation of helicopter I above the macro-terrain, as recorded in COMMON/ ELOCZ/ (see Volume 2).

This difference in method of computing elevations for helicopters means that in each instance within DYNCOM that an elevation for a vehicle is desired, it is necessary to determine whether or not the vehicle is a helicopter and then to apply the proper relationship for elevation. However, the approach that has been taken has simplified the task and has permitted incorporation of the modification throughout the simulation without an extensive amount of reprogramming.

Determining Elevations

The function, ELVATE, has been modified to yield an elevation as follows:

Z = ELVATE(X, Y, IC)

where X, Y are the coordinates of the position on the battlefield for which an elevation Z is desired, and IC is the element number of the vehicle occupying the position X, Y. The convention is followed that IC = 0 indicates that the terrain elevation of the point X, Y is desired.

This approach does require that each call to function, ELVATE, be modified to include the new calling parameter. However, logic external to ELVATE to determine a value for IC has been rarely required since it is always known

whether an elevation for a vehicle or a point on the ground is desired. Also, if a vehicle elevation is desired, the element number is usually known. Moreover, the approach allows the continued usage of the two original relationships

Z = HV + EM + HTER for observer and target vehicles

and

Z = HTER for terrain elevations.

Thus, additional logic external to function, ELVATE, is not required. The reason that the original relationships still hold is that EM for helicopters is always recorded as zero and HTER comes from the modified version of function ELVATE.

The processing sequence for the revised version of function ELVATE will now be given. This function appears in flow chart form in Volume 2.

1. Determine the terrain elevation at the position specified by the calling parameters X, Y; i.e., set Z = HTER.

2. If the elevation desired is for a point on the ground (IC = 0), go to step 6.

3. Determine the helicopter number of the vehicle at X, Y; i.e., set

   L = LHICE(IC)

   where the array LHICE is contained in COMMON/LHICE/.

4. If the vehicle is not a helicopter (L = 0), go to step 6.

5. Increment the elevation by the recorded elevation of the helicopter above the terrain; i.e.,

   $Z + ELOCZ(L) \rightarrow Z.$

6. The computational sequence is complete.

Fire Controller Revisions

The original version of DYNCOM viewed MISTIC missile launchers and MISTIC forward observers as regular vehicular elements with dual roles in combat. The firing activities of a MISTIC launcher were normally controlled by

input data so that such an element would select a direct-fire target for engagement only when the target posed a threat to the survival of the launcher. The primary role of the launcher was to respond to requests from MISTIC forward observers for indirect MISTIC fire. These fire control procedures were contained entirely in the fire controller for ground based direct-fire elements programmed as subroutine FIRCON. Futhermore, the firing activities of a MISTIC forward observer (FO) were controlled in a manner similar to that employed for MISTIC launchers. The primary function of the FO was to select targets for indirect-fire attack and to illuminate such targets when indirect-fire was delivered. The FO's chose direct-fire targets for engagement only in self defense and again, all the fire control procedures were performed in subroutine FIRCON.

In DYNCOM these same fire control philosophies exist for MISTIC launchers and for all FO's. However, the indirect-fire activities are now controlled by the model reported in reference 2 for FO's and the model reported in Chapter 3 for launchers. Thus, changes have been required in the fire controller as implemented in subroutine FIRCON. Only those activities of launchers and FO's which are related to selection and engagement of direct-fire targets are now represented in the fire controller.

The changes cited above have, for the most part, involved the removal of a substantial portion of logic from subroutine FIRCON. One section of removed logic was associated with selection of an indirect-fire target by an FO, addition of the target to his fire request list, and communication of a fire request. It was also involved with subsequent deletion of the target if the target became undesirable for attack. The logic now appears in revised form in subroutine AFO.

Another section of removed logic was associated with selection by a MISTIC launcher of an indirect-fire target from the fire mission list. The logic was also involved with determining when indirect fire could take place. All this launcher logic now appears in revised form in subroutine LAUNCH and MFB, described in Chapter 3.

There have been procedures that have been added to subroutine FIRCON, however. Some of these procedures are required to properly control the direct-fire activities of launcher and FO vehicles when they are otherwise involved with an indirect-fire mission. Other procedures are involved with controlling the indirect-fire activities of the MISTIC launcher and FO elements when the vehicles with which they are associated are engaging direct-fire targets. We will outline each procedure in the paragraphs which follow.

Limits on Direct-Fire Activities

Direct-fire activities of MISTIC launcher and FO vehicles consist of selecting a target for attack, selecting and loading ammunition to be used, and

firing the projectile. The fire controller (subroutine FIRCON) represents the target and ammunition selection processes and determines whether the vehicle must stop in order to fire. Subroutine FIRMOD, LAUNCH or SHILLY (depending upon the projectile type selected) represents the loading and firing activities. Note that a MISTIC launcher may select a conventional round for direct-fire or it may select a MISTIC missile for direct-fire. The ammunition selection process has been described in reference 3. Note too that the missile launcher fire support element may continue communication activities for indirect fire during a direct-fire mission (see Chapter 3).

Launcher Without Assignments

In subroutine FIRCON, a MISTIC launcher vehicle without a direct-fire target is not allowed to select a target if the associated launcher element is presently involved in an indirect-fire mission. A direct-fire assignment is not permitted in this situation, and the vehicle is not processed for target selection.

To determine whether or not the launcher element is involved with an indirect-fire mission, subroutine FIRCON examines the variable IFBMIS(NF), where the subscript NF is the firer number of the launcher element as defined in Chapter 1. The variable, IFBMIS(NF) > 0, indicates that the launcher element is engaged in some phase of an indirect-fire mission, since the launcher is considered to be "involved" if IFBMIS(NF) > 0, a direct-fire assignment is not allowed.

Launchers With In-Flight Missiles

If a MISTIC launcher already has a direct-fire target, it may be that the launcher has previously fired a MISTIC round at the target. In this situation, it may be that the round is still flying at the time that the launcher vehicle is processed by subroutine FIRCON. If this is the case, the present firing assignment must be continued. That is, it cannot be deleted and no attempt can be made to select another target. Therefore, no processing is required in FIRCON until the round that is flying has impacted. This will occur on a subsequent event of the launcher vehicle.

To mechanize the process discussed above, the variable, LFLAG(NUM) is used where NUM is the launcher number of the launcher vehicle. The array, LFLAG, is stored in COMMON/LFLAG/. When an assignment is made and a MISTIC missile is launched, LFLAG(NUM) is set to two. The variable is not reset to zero until the missile impacts or its flight is aborted. The reset is accomplished in the MISTIC flight routine FLIGHT. The time of impact or abort, TDFRDY(NUM), is also set in subroutine FLIGHT, where TDFRDY is contained in COMMON/TDFRDY/. Therefore, the launcher vehicle cannot fire a MISTIC missile at a direct-fire target if

$$LFLAG(NUM) > 0$$

or if

$$LFLAG(NUM) = 0$$

and

$$CLOCK \leq TDFRDY(NUM)$$

where CLOCK is the clock time of the launcher vehicle.

Launcher and FO Firing Thresholds

Since MISTIC FO's and launchers have a primary role in indirect-fire activities and normally engage direct-fire targets only in self defense, some method is required to restrict the direct-fire targets selected in subroutine FIRCON to be those posing a sizable threat. The same is true of the other types of FO's discussed in reference 2. The method used is summarized below.

Within FIRCON, a direct-fire target is selected on the basis of adjusted target range, RAF. Each potential target element is assigned an adjusted range and the element with the smallest adjusted range is selected as the target. The adjusted range is actually the sum of multiple factors related to the desirability of the potential target element. Normally, the process commences by initializing the assigned target indicator, KTGT, to zero and by initializing the smallest recorded adjusted range, ADJR, to a very large number (∞). Then each enemy element I is analyzed, its adjusted range, RAF, is computed, and if RAF < ADJR, the element is recorded as the assigned target; i.e.,

$$I \rightarrow KTGT$$

$$RAF \rightarrow ADJR.$$

When all elements have been analyzed, the target, KTGT, is assigned for direct-fire. If KTGT = 0, no elements qualify for assignment.

For MISTIC launcher vehicles and FO vehicles, the procedure is slightly different. The initial value of ADJR is set to a threshold range instead of to a very large number. In this way, the target selected is controlled to be one that actually poses a threat which exceeds the threshold implied by the initial value of ADJR.

For launchers, the threshold range is contained in the array RAFMNL stored in COMMON/RAFMNL/. The threshold range is found from the array as follows:

ADJR = RAFMNL(LCOD)

where LCOD is the MISTIC unit weapon code of the launcher as defined in Chapter 1; i.e.,

LCOD = LWCOD(NSUB) - MWART

where

NSUB = NUMELE + NUMART + NM

and

NM = NMISUN(NUM).

Here, NUM is the launcher number of the vehicle and NMISUN(NUM) is the MISTIC unit to which NUM belongs. Other defining constants are contained in COMMON/NUMBER/ (see Volume 2 for all COMMON's mentioned).

For MISTIC FO's (and other FO's also) the threshold range is found from the array, RAFMNF, stored in COMMON/RAFMNF. The threshold range is ADJR = RAFMNF(I, KP1)

$$\text{where KP1} = \begin{cases} 1, \text{ if the FO belongs to the blue force} \\ 2, \text{ if the FO belongs to the red force} \end{cases}$$

$$\text{and I} = \begin{cases} 1, \text{ if the FO is an artillery FO} \\ 2, \text{ if the FO is a MISTIC FO} \\ 3, \text{ if the FO is a special FO.} \end{cases}$$

The three types of FO's discussed above are described in reference 2. There also, will be found a description of the technique for determining what type of FO is being analyzed.

Restricted Movement

The firing activities of vehicles sometimes affect their movement. That is, vehicles with targets are sometimes unable to move and fire simultaneously. Therefore, if an element in a section has a direct-fire target and cannot fire while moving, the section must be brought to a stop. DYNCOM is designed to represent this phenomenon. However, the only processing required in FIRCON is to determine whether or not fire and movement by a vehicle are allowed, and if not, to set flags that will result in the element's section being brought to a stop.

For all vehicles I with direct-fire assignments, the criterion that allows fire while moving is that the range to the direct-fire target MDFAF(I) be less than an input threshold range. This range is obtained from the array, RMVFR, stored in COMMON/RMVFR/. The process for determining the correct value from the array is described in reference 4 and will not be repeated here.

Even when a vehicle does not have a direct-fire target (MDFAF(I) = 0), it may be that it still cannot move. This is true when the vehicle is a launcher or is carrying a forward observer team and is engaged in an indirect-fire mission.

The reader will recall that a launcher is engaged in an indirect-fire mission if IFBMIS(NF) > 0, where NF is the fire support firer number of the launcher. If the launcher has a mission, the model assumes that the vehicle must stop. That is, within DYNCOM, launchers are not allowed to fire indirect- fire missions while moving.

For FO's, DYNCOM uses the convention that movement is restricted when illumination of a target is being conducted and the user has specified that movement during illumination is not allowed. The user initializes the array IFOMI (COMMON/IFOMI/) to implement the convention; i.e.,

$$\text{IFOMI(NUM)} = \begin{cases} 1, \text{ if the FO NUM can move while illuminating} \\ 0, \text{ if otherwise} \end{cases}$$

for each FO NUM in the battle.

To determine whether an FO is currently illuminating a target, subroutine FIRCON determines whether a missile which the FO requested has been launched. It is assumed that the FO almost constantly illuminates from the time that the first missile requested against a target complex is launched until the time that the last missile requested impacts.

The procedure to determine whether illumination is underway consists of searching through the appropriate set (red or blue) of MISTIC missile clocks I to determine whether any missiles are presently active (ECLOCK(I) $< \infty$). For each active missile, the FO requesting the missile is found from the proper entry in the array, MISAVE. If this FO corresponds to the FO being processed by subroutine FIRCON, then illumination is underway.

Launcher Ammunition Supplies

In subroutine FIRCON a necessary condition for continuation of an existing direct-fire assignment is that there remain at least one round of the ammunition that was selected for the engagement when the assignment was made. If an ammunition supply is depleted, the assignment is deleted.

For all elements I the remaining ammunition supply of ammunition type J is contained in the array LAMMO(J, I). Therefore, an assignment for element I, using ammunition type J, can continue if LAMMO(J, I) > 0. However, if the element is a launcher, it may be that the assignment can be continued even if (LAMMO(J, I) = 0. This is true when the ammunition that is being fired is a MISTIC missile. In this case, LAMMO(J, I) represents the number of missiles loaded and ready to fire. It is assumed that there is an additional supply, NM(NUM), stowed aboard the launcher, NUM, that could be loaded. Thus, an assignment using MISTIC missiles is deleted only when

LAMMO(J, I) + NM(NUM) = 0.

Of course, J must be the MISTIC missile ammunition; i.e., J must be equal to IFMC(LCOD) where LCOD is the MISTIC unit weapon code of the launcher as defined previously, and IFMC(LCOD) is the ammunition code of the MISTIC missile. The arrays, LAMMO, NM and IFMC, are contained in common areas bearing the array names and must be initialized.

FO's With In-Flight Missiles

One final restriction applies to the direct-fire activities of FOs that are engaging indirect-fire targets. It is assumed that a vehicle carrying an FO team cannot fire at a direct-fire target if the FO is currently illuminating targets for an indirect-fire attack. Thus, if a direct-fire assignment exists, the

actual firing event will be delayed until all missiles which the FO requested have impacted and the FO is no longer illuminating. The procedure used to determine whether the FO is illuminating is exactly that outlined previously in our discussion of restricted movement of FO vehicles.

Limits on Indirect-Fire Activities

As the last operation in subroutine FIRCON, forward observer vehicles and MISTIC launcher vehicles are processed by subroutine ARFO to determine any restrictions that apply to the fire support activities of the FO teams and launcher elements as a result of the direct-fire activities of the transporting vehicles. If restrictions apply, flags are set so that the FO and the launcher elements will recognize that their fire support activities have been limited. If restrictions do not apply, processing is performed to permit the fire support elements to resume their fire support activities. Understanding of the material outlined below can be augmented by the more complete discriptions of FO and launcher activities presented in reference 2 and Chapter 3, respectively.

The flag that is used to control the fire support activities of launchers and FOs is contained in the array, IFRFL(I). This array has one entry for each FO element being represented (I = 1,..., ITOTFO) and one entry for each launcher (I = ITOTFO + 1,..., ITOTFO + ITOTLN). From the standpoint of subroutine ARFO, the entries are defined as follows:

$$\text{IFRFL(I)} = \begin{cases} \text{1, if all fire support activities of the launcher or} \\ \quad \text{FO are suspended} \\ \text{0, if otherwise.} \end{cases}$$

Actually, other values are permitted for launchers as explained in Chapter 3. However, only the two values indicated are of interest here.

Launcher Processing

The processing required for launchers is quite simple. If the launcher vehicle I is engaging a target with direct-fire, (MDFAF(I) > 0), the fire support activities are suspended. Thus, in this case, IFRFL(KSUB) is set to one where KSUB = ITOTFO + NUM and NUM is the corresponding launcher element number. If the launcher vehicle is not engaging a direct-fire target, (MDFAF(I) = 0), the launcher element is returned to fire support activities if required. That is, if IFRFL(KSUB) = 1, then IFRFL(KSUB) is set to zero and the clock of the launcher element is reset so that the element will become current. The clock is set by the procedure

ECLOCK(NFBCLK) = CLOCK + EPSILN

where

NFBCLK = launcher element clock number

CLOCK = present clock time of the launcher vehicle (COMMON/ICECOM/)

EPSILN = small positive number (see COMMON/SEQPAR/).

FO Processing

The forward observer fire support activities are controlled in a manner similar to that used for controlling launcher activities. However, the conditions that result in interrupted fire-support activities are different. First, if the FO vehicle is neutralized, it is assumed that the FO team cannot select fire-support targets. To determine whether the vehicle is neutralized, entries in the array TNEUT are used. From the definition of TNEUT, neutralization of vehicle I exists if

$$\text{CLOCK} < \text{TNEUT}(1, I)$$

or

$$\text{TNEUT}(2, I) < \text{CLOCK} < \text{TNEUT}(3, I)$$

where CLOCK is the present clock time of vehicle I.

Next, if the vehicle is not neutralized, the fire-support activities are interrupted if a direct-fire target exists (MDFAF(I) > 0) and the vehicle is to fire during the event. The vehicle is to fire if IFLE > 0 where IFLE is the firing event flag recorded by subroutine FIRCON in COMMON/ICECOM/.

Finally, a special procedure is used if an FO team is not assigned to an artillery or a MISTIC unit. Such teams are defined in reference 2 as special FO teams and represent elements such as platoon leaders whose normal function is to engage targets with direct fire. These elements only act as fire-support elements when the enemy threat is sufficient to cause them to call for fire support. At other times their fire-support activities are suspended.

A measure of the threat for FO NUM is contained in the intense fire fight flag, NSTHFF(NUM), defined as follows:

$$\text{NSTHFF(NUM)} = \begin{cases} 1, & \text{if an intense fire fight exists} \\ 0, & \text{if otherwise} \end{cases}$$

The array, NSTHFF, contains one entry for each FO team being represented and is evaluated as explained in a paragraph to follow. Thus, the fire support activities of a special FO team are suspended if an intense fire fight does not exist; i.e., if NSTHFF(NUM) = 0.

The intense fire fight flag for FO NUM is set to one (fire fight exists) in subroutine ARFO if

$$FFRAT > CF(K)$$

where FFRAT is the ratio of the number of known enemy elements that are presently firing to the number of surviving friendly elements, and CF(K) is the input threat ratio for an FO of type K. The subscript, K, is defined as follows:

$$K = \begin{cases} 1, & \text{if a blue FO is assigned to an artillery unit} \\ 2, & \text{if a blue FO is assigned to a MISTIC unit} \\ 3, & \text{if a blue FO is a special FO} \\ 4, & \text{if a red FO is assigned to an artillery unit} \\ 5, & \text{if a red FO is assigned to a MISTIC unit} \\ 6, & \text{if a red FO is a special FO.} \end{cases}$$

The ratio, FFRAT, is determined in subroutine ISTHFF.

VARIABLE DEFINITION INDEX

REFERENCES

1. Clark, G. M., et al., Small Unit Combat Simulation (DYNTACS X), Air Defense Operations Models, RF 2978 FR 71-2A (U), edited by G. M. Clark, S. H. Parry, D. C. Hutcherson, and J. J. Rheinfrank III, Systems Research Group, The Ohio State University, Columbus, O., March 1971, AD 887265L.

2. Hutcherson, D. C., "Fire Support Target Selection and Coordination," Chapter 2 (in) Small Unit Combat Simulation (DYNTACS X), Fire Support Operations Models, RF 2978 FR 71-3A (U), edited by G. M. Clark and D. C. Hutcherson, Systems Research Group, The Ohio State University, Columbus, Ohio, October 1971, AD 890443.

3. Clark, G. M., "Movement and Firing Tactics," Chapter 2 (in) The Land Combat Model (DYNCOM), RF 2376 FR-1 (U), edited by A. B. Bishop and G. M. Clark, Systems Research Group, The Ohio State University, Columbus, Ohio, June 1969, AD 909540.

4. Clark, G. M., "Fire Controller," Chapter 6 (in) The Tank Weapon System, edited by A. B. Bishop and G. M. Clark, RF 573 AR 69-2B (U), Systems Research Group, The Ohio State University, September 1969, AD 864920.